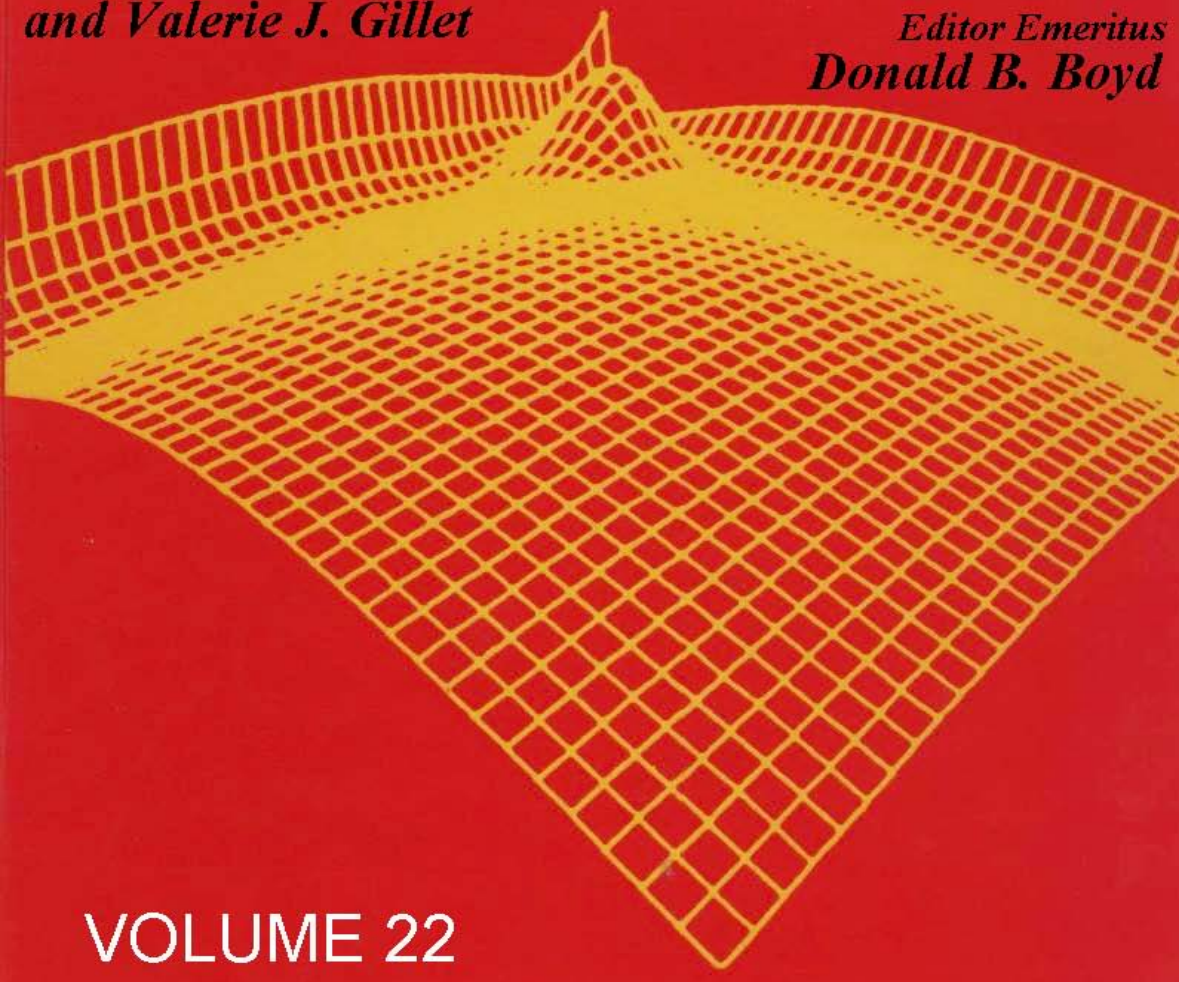
REVIEWS IN COMPUTATIONAL CHEMISTRY

Kenny B. Lipkowitz, Thomas R. Cundari, and Valerie J. Gillet

Editor Emeritus
Donald B. Boyd



VOLUME 22

WILEY-VCH

Reviews in Computational Chemistry Volume 22

Reviews in Computational Chemistry Volume 22

Edited by

Kenny B. Lipkowitz, Thomas R. Cundari, and Valerie J. Gillet

Editor Emeritus

Donald B. Boyd

WILEY-VCH

Kenny B. Lipkowitz
Department of Chemistry
Howard University
525 College Street, N.W.
Washington, D. C., 20059
ken.lipkowitz@cox.net

Thomas R. Cundari
Department of Chemistry
University of North Texas
Box 305070,
Denton, Texas 76203-5070, U.S.A.
tomc@unt.edu

Valerie J. Gillet
Department of Information Studies
University of Sheffield
Regent Court, 211 Portobello Street
Sheffield, S1 4DP U.K.
v.gillet@sheffield.ac.uk

Donald B. Boyd
Department of Chemistry
Indiana University-Purdue University
at Indianapolis
402 North Blackford Street
Indianapolis, Indiana 46202-3274, U.S.A.
boyd@chem.iupui.edu

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

ISBN-13 978-0-471-77938-4
ISBN-10 0-471-77938-5
ISSN 1069-3599

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Preface

Toward the end of the twentieth century, a series of well-planned and visionary conferences, along with successful developments in both scientific achievement and policy making, led to a 1988 memorandum of interagency cooperation that provided the foundation for an NIH-DOE collaboration to achieve the goals of the U.S. Human Genome Project (HGP) (Major Events in the U.S. Human Genome Project and Related Projects: http://www.ornl.gov/sci/techresources/Human_Genome/project/timeline.shtml). What followed was a momentous confluence of talent, ego, finances, and hard work dedicated to determining all genes, now estimated at 20,000–25,000 in number, from all three billion base pairs in the human genome. It was a project of epic proportion; tens of organizations, hundreds of laboratories, and thousands of workers eventually achieved that goal and reported their work, formally, by concurrent publications in mid-February of 2001 (free online publications can be found at http://www.nature.com/genomics/index.html and http://www.sciencemag.org/content/vol291/issue5507/). The HGP was completed in 2003.

As the frenetic pace of genomics quickened near the turn of the century, most of us not involved in that fray were cognizant that another, more valuable prize, the human proteome, was being targeted even as concrete was being poured for buildings to house new departments, institutes, and companies dedicated to genomic research. Of the major classes of biological molecules, proteins have had the scientific spotlight focused on them in the past, and they will continue to enjoy that spotlight shine for the foreseeable future. The significance of proteins, from the perspective of basic science where curiosity-driven exploration takes place to industry where economic engines drive advances in medicine, is unrivaled and is a focus of this, the twenty-second volume of *Reviews in Computational Chemistry*.

One project that will advance our understanding of the proteome is the Protein Structure Initiative (PSI: http://www.nigms.nih.gov/psi/). Its goal is "... to make the three-dimensional atomic-level structures of most proteins easily obtainable from knowledge of their corresponding DNA sequences." Here, high-throughput protein structure generation is taking place on an unprecedented scale to achieve a systematic sampling of major protein

families. How can one distill all of these data into something that is useful? One way is to rely on classification, one of the most basic activities in all scientific disciplines. It is easier to think about a few groups that share something in common than it is to think about each individual, and since the first scientific classification by Aristotle in the fourth century B.C., through the binomial system of nomenclature by Linnaeus in the eighteenth century, and continuing to the classification of protein structure/function in modern structural biology, it is clear that the wealth of information available, especially from genome sequencing projects, is best studied through classification in its broadest sense.

In Chapter 1, Professor Patrice Koehl focuses on the little recognized, albeit significant, topic of protein structure classification. In this tutorial, the author first describes proteins and then surveys their different levels of organization, from their primary structure (sequence) through their quaternary structure in cells. Protein building blocks, structure hierarchy, types of proteins, and protein domains are defined and explained for the beginner. Links to online resources related to protein structure and function are provided. The crux of this tutorial is on protein structure comparison and classification. Described in detail are computational methods needed for automatically detecting domains in protein structures, techniques for finding optimal alignment between those domains, and new developments that rely on the topology of the domain rather than on its structure. This is followed by a review of protein structure classification. Proteins are first divided into discrete, globular domains that are then further classified at the levels of class, folds, superfamilies, and then families. After reviewing the terms that define a classification, the three main protein structure classifications, SCOP, CATH, and the DALI Domain Dictionary, are then described and compared. Resources and links to these and other methods are given. The ability to organize the existing, voluminous data related to protein structure and function in a way that evolutionary relationships can be uncovered, and to detect remote homologues in the rapidly developing area of structural biology, is emphasized in this chapter. The author provides tables of resources related to protein structure and websites containing publicly available services and/or programs for domain assignment and structure alignment. Also provided are databases of protein structural domains and resources for protein sequence/protein structure classification. In the burgeoning field of structural biology exemplified by the PSI, these techniques and tools are necessary for advancement and the author provides a complete tutorial/review of the techniques and methodologies needed for protein structure classification.

Given that elegant advances are being made in automated protein structure classification and even with the soon-to-be-initiated production stage of the PSI (called PSI-2), the difficulties inherent in protein crystallization imply that not all possible protein structures will be known in the near future. Accordingly, there is a need to predict at atomic resolution the three-dimensional (3-D) shape of novel "designer" proteins and proteins whose

sequence is known, but for which no crystal structure is available. The following two chapters on the topics of homology modeling and simulations of protein folding address the history, the needs, and the many advances that have been made in determining structures of proteins computationally.

In Chapter 2, Drs. Emilio Esposito, Dror Tobi, and Jeffry Madura provide a tutorial on the topic of comparative protein modeling, a.k.a., homology modeling. Although many proteins from similar families have similar functions, it is common to find instances where proteins with similar structures have different functions. The authors describe in this chapter how to first construct a protein structure and then how to validate its quality as a model. The first step in homology modeling is to search for known, related sequences and structures by using, for example, the Protein Data Bank (PDB), or the Expert Protein Analysis System (ExPaSy) website, which contains useful databases like SWISS-PROT, PROSITE, ENZYME, and SWISS-MODEL. Details about these databases along with pitfalls to avoid when using them are provided. The next step, which is most critical in a comparative modeling study, is sequence alignment. Both global, coarse-grained alignment strategies and local, fine-grained alignment strategies are described. The basics of alignment are given for the novice modeler, insights about sequence preparation are passed on to the reader, and common alignment tools like BLAST, Clustal (and their progeny), T-Coffee, and Divide-and-Conquer are described. The differences between progressive and fragment-based methodologies are highlighted, and a description about how one scores the final alignment to select the best model is given. The next two steps in homology modeling involve template selection and improving alignments. Methods like threading and uses of hydropathy plots are described before a tutorial is presented on how to actually construct a protein model. The difference between finding the best model versus a consensus model is highlighted, as is the need for satisfying spatial constraints. Segment match modeling, multiple template methods, hidden Markov modeling, and other techniques are identified and explained for the novice. The penultimate step of refining the protein structures using, e.g., databases like Side-Chains with Rotamer Library (SCWRL) or by implementing atomistic simulation methods like simulated annealing is then described. Finally, the authors inform us about how to evaluate the validity of the derived protein structures using PROCHECK, Verify3D, ProSa, and PROVE in addition to existing tools from the realm of spectroscopy such as found in the OLERADO suite of applications. For each step of the homology modeling process, they provide a working example to illustrate some problems and pitfalls a novice could encounter, and they provide tables of key websites containing databases and computational resources needed for homology modeling.

In a 1992 publication entitled "One Thousand Families for the Molecular Biologist," (*Nature*, 1992: 357, 543), Cyrus Leventhal estimated that for the native state of a single domain protein, approximately 1000 different shapes or folds exist in nature. Although that assertion may be true, the

most recent assessment of protein fold space by Hou, Sims, Zhang, and Kim (*Proceedings of the National Academy of Sciences*, 2003; 100(5): 2386, available online for free at http://www.pnas.org/content/vol102/issue10/) confirms the notion that the "protein fold space" is not homogeneous but is, instead, populated in a highly nonuniform manner. Using one domain structure from each of the 498 SCOP folds, a pair-wise structural alignment was carried out by those authors leading to a 498 × 498 matrix of similarity scores. Then, using distance–geometry concepts, a distance matrix was generated that was thereafter transformed into a metric matrix, the eigenvalues of which are orthogonal axes passing through the geometric centroid of the points representing the folds. The three dominant eigenvalues are shown in Figure 1 and reveal several interesting features of protein fold space, the most important of which is that the α, β, and α/β folds are clustered around three separate axes, whereas the α + β folds lie approximately on a plane formed by two of those axes.

The take-home message from this assessment is that proteins with varying numbers and patterns of amino acids adopt similar 3-D shapes; the emptiness of protein fold space is most likely attributable to the finding that many protein shapes are architecturally unstable. Even with this knowledge, it is still

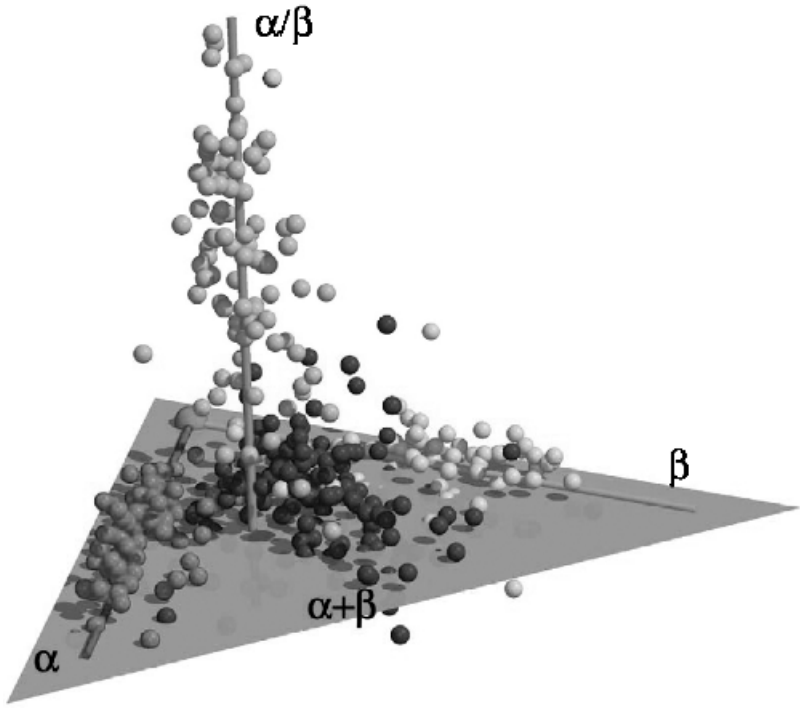


Figure 1 The 3-D representation illustrates the clustering of structures along separate axes and highlights obvious voids in protein fold space. (Reproduced with permission from *PNAS*, 2003; 100(5): 2386.)

not possible to predict, either quickly or accurately, the shape of a folded protein given only the sequence of its constituent amino acids. Understanding the factors that contribute to folding rates and thermodynamic stability is thus crucial for delineating the folding process.

In Chapter 3, Professor Joan-Emma Shea, Ms. Miriam Friedel, and Dr. Andrij Baumketner present a tutorial on protein folding simulations, the aim of which is not only directed toward helping a modeler predict a protein's shape but also toward revealing, for the novice, the theoretical underpinnings of why and how that shape exists, especially when compared with other heteropolymers that do not fold into a well-defined ground-state structure. The authors begin by examining the Levinthal paradox, which states that if a protein had to search randomly through all of its possible conformational states to reach the native state, the folding time would be prohibitively long—on the order of the lifetime of the universe for moderately sized systems. They then introduce energy landscape theory, whose foundation is built on the concept of frustration in spin glass systems, along with earlier models that explain the folding process, including diffusion–collision, hydrophobic collapse, and nucleation models. The thermodynamics and kinetics of folding is then presented, and connections with experimental observations are made. Most of the tutorial/review covers general simulation techniques. The authors begin with the coarse-grained modeling techniques of lattice and off-lattice models, the former of which are typically performed with Monte Carlo searches with simplified representations of the constituent amino acids required to remain on a lattice, whereas the latter are performed with Langevan and discontinuous molecular dynamics methods in which the simplified amino acid components are allowed to move in continuous space. The history, methodology, advantages, and disadvantages of these techniques are presented in a straightforward way for the beginning modeler. This introduction is followed by a discourse on fully atomistic models. After a brief introduction about force fields and their uses, the authors describe the stochastic difference equation (SDE) method, caution the reader about relying too heavily on the principle of microscopic reversibility (so that one is not tempted to use unfolding trajectories to infer the folding mechanism), and describe importance sampling to generate free energy surfaces for folding. This part of their tutorial ends with a description of replica-exchange as an increasingly attractive and tractable means to study the thermodynamics of folding. The final portion of the chapter focuses on the transition state ensemble (TSE) for folding. Transition state and two-state kinetics are introduced. Methods for identifying the TSE including reaction coordinate-based methods, nonreaction coordinate-based methods, and φ-value analysis are introduced briefly, explained in a cogent manner, and then reviewed thoroughly. Ongoing developments in this area of protein science are described, and future directions for advancements are identified.

In Chapter 4, Marco Saraniti, Dr. Shela Aboud, and Robert Eisenberg introduce the mathematics and biophysics of simulating ion transport through biological channels. Understanding how ion channels work has become a hot

and very controversial area of research in the past three years in part because of the limitations of discerning molecular motions from X-ray crystallographic studies—a situation in which simulation can help clarify many controversies. This chapter is an introduction to the numerical techniques used for such simulations. The authors begin by first describing the types of proteins involved, providing as specific examples Gramicidin A and Porins. They then describe the membrane consisting of its amphiphilic lipid molecules and attendant molecules like steroids, provide insights about how best to treat the aqueous environment, and finally they demonstrate how all of these constituents must be assembled to represent the full system being modeled. Because ensemble and time averages are being computed for comparison with experiment, the authors then focus on the time scales and space scales involved and emphasize that one hallmark of this type of protein modeling is that measurable quantities of direct biological interest evolve in time up to 12 orders of magnitude, from femtoseconds to milliseconds. The electrostatic treatments used in computing the long-range interactions is then described in an easy-to-follow tutorial that covers the fast multipole method (FMM), Ewald summation methods, solving Poissons's equation in real space, finite difference iterative schemes, and the uses of multigrid methods. Error reduction in classic iterative methods is presented, and a minitutorial on multigrid basics is given for the novice. A description of how one treats the short-range forces and boundary conditions is then presented before the authors describe particle-based simulation strategies. Both implicit and explicit treatments of solvent are covered. In the former treatment, the Langevin formalism with its temporal discretization and the associated integration schemes needed for such Brownian dynamics simulations are described. In the latter treatment, the water models used in Newtonian dynamics are described. Because these particle-based simulation methods are limited to small spatial scales and short time periods, the authors then devote an entire section of their tutorial to flux-based, (i.e., electrodiffusive) methods, in which current densities flowing through the system can be treated on biologically relevant time and size scales. The Nernst–Planck equation is described in detail, and then the Poisson–Nernst–Planck (PNP) method is introduced; the novice is guided, step-by-step, through the processes needed for a successful simulation, with simple illustrations and easy-to-follow equations. Flux-based methods belong to the family of continuum theories of electrolytes that are based on the mean field approximation. The advantages, disadvantages, assumptions, and approximations of these continuum methods are given in a straightforward way by the authors along with insights about what one can do and cannot do with such computational techniques. The hierarchy of simulation schemes needed to obviate problems with scales of time and space are presented clearly in this tutorial/review.

The final chapter of this volume covers the topic of wavelet transforms, a general technique that can be used in protein-related research as well as for a multitude of other needs in computational chemistry, informatics, engineering,

and biology. In Chapter 5, Professors Curt Breneman and Mark Embrechts review the topic of wavelets in chemistry and chemical informatics with their students C. Matthew Sundling, Nagamani Sukumar, and Hongmei Zhang. Unlike traditional signal processing methods, the wavelet transform offers simultaneous localization of information in both frequency and time (or property) domains and is well suited to processing data containing complex and irregular property distributions or waveforms into simple, yet meaningful components. The method developed quickly in the 1990s with many applications in spectroscopy, chemometrics, quantum chemistry, and more recently chemoinformatics. This pedagogically driven review begins with an introduction to wavelets. The Fourier transform, continuous- and short-time Fourier transforms, are described with simple mathematics as are the wavelet transform, the continuous-, discrete-, and wavelet packet transforms. The chapter is replete with illustrations describing the concepts and the mathematics associated with each technique. After this tutorial the authors provide examples of wavelet applications in chemistry with an emphasis on smoothing and denoising, signal feature isolation, signal compression, and quantum chemistry. Their chapter ends with a survey of how wavelets are used in classification, regression, and QSAR/QSPR. The authors provide a simple tutorial for the novice molecular modeler and create a compelling rationale for why wavelets are so useful to computational scientists in chemistry and informatics.

We are delighted to report that the Institute for Scientific information, Inc. (ISI) rates the *Reviews in Computational Chemistry* book series in the top 10 in the category of ''general'' journals and periodicals. The reason for these accomplishments rests firmly on the shoulders of the authors whom we have contacted to provide the pedagogically driven reviews that have made this ongoing book series so popular. To those authors we are especially grateful.

We are also glad to note that our publisher has plans to make our most recent volumes available in an online form through Wiley InterScience. Please check the Web (http://www.interscience.wiley.com/onlinebooks) or contact reference@wiley.com for the latest information. For readers who appreciate the permanence and convenience of bound books, these will, of course, continue.

We thank the authors of this and previous volumes for their excellent chapters.

Kenny B. Lipkowitz
Washington

Valerie J. Gillet
Sheffield

Thomas R. Cundari
Denton

July 2005

Contents

1. Protein Structure Classification **1**
Patrice Koehl

Introduction 1
Classification and Biology 2
The Biomolecular Revolution 2
Basic Principles of Protein Structure 4
Visualization 4
Protein Building Blocks 5
Protein Structure Hierarchy 5
Three Types of Proteins 7
Geometry of Globular Proteins 9
Protein Domains 12
Resources on Protein Structures 13
Protein Structure Comparison 14
Automatic Identification of Protein Structural Domains 14
The Rigid-Body Transformation Problem 16
Protein Structure Superposition 23
cRMS: An Ambiguous Measure of Similarity 31
Differential Geometry and Protein Structure Comparison 33
Upcoming Challenges for Protein Structure Comparison 36
Protein Structure Classification 37
The Structure Classification of Proteins (SCOP) 40
The CATH Classification 42
The DALI Domain Dictionary (DDD) 43
Comparing SCOP, CATH, and DDD 43
Conclusions 45
Acknowledgments 46
Appendix 47
References 48

2. Comparative Protein Modeling 57
Emilio Xavier Esposito, Dror Tobi, and Jeffry D. Madura

Introduction 57
Anatomy of a Comparative Model 60
Step 1: Searching for Related Sequences and Structures 61
Expert Protein Analysis System (ExPASy) 62
BLAST and PSI-BLAST 65
Protein Data Bank (PDB) 68
Sequence Alignment and Modeling System with Hidden Markov Models 70
Threading 73
Threader 78
Example: Finding Related Sequences and 3-D Structures 80
Step 2: Sequence Alignment 84
Preparing the Sequences 87
Alignment Basics 90
Similarity Matrices 91
Clustal 95
Tree-Based Consistency Objective Function for Alignment Evaluation (T-Coffee) 99
Divide-and-Conquer Alignment (DCA) 100
Example: Aligning Sequences 101
Step 3: Selecting Templates and Improving Alignments 104
Selecting Templates 104
Improving Sequence Alignments With Primary and Secondary Structure Analysis 107
Example: Aligning the Target to the Selected Template 111
Step 4: Constructing Protein Models 111
Satisfaction of Spatial Restraints 113
Segment Match Modeling 115
Multiple Template Method 118
3D-JIGSAW 119
Overall Protein Model Construction Methods 121
Example: Constructing a Protein Model 122
Step 5: Refinement of Protein Models 124
Side-Chains with Rotamer Library (SCWRL) 125
Energy Minimization 132
Molecular Dynamics 133
Molecular Dynamics with Simulated Annealing 135
Step 6: Evaluating Protein Models 138
PROCHECK 138
Verify3D 140

ERRAT 141
Protein Structure Analysis (ProSa) 142
Protein Volume Evaluation (PROVE) 144
Model Clustering Analysis 146
Example: Evaluation of Protein Models 148
Conclusions 154
References 155

3. Simulations of Protein Folding 169
Joan-Emma Shea, Miriam R. Friedel, and Andrij Baumketner

Introduction 169
Theoretical Framework 172
Energy Landscape Theory 172
Thermodynamics and Kinetics of Folding: Two-State and Multistate Folders 175
Protein Models 179
Introduction and General Simulation Techniques 179
Coarse-Grained Protein Models 181
Fully Atomic Simulations 190
Advanced Topics: The Transition State Ensemble for Folding 201
Transition State and Two-State Kinetics 202
Methods for Identifying the TSE 204
Conclusions and Future Directions 219
Acknowledgments 219
References 220

4. The Simulation of Ionic Charge Transport in Biological Ion Channels: An Introduction to Numerical Methods 229
Marco Saraniti, Shela Aboud, and Robert Eisenberg

Introduction 229
System Components 231
Time and Space Scale 241
Experiments 242
Electrostatics 243
Long-Range Interaction 245
Short-Range Interaction 258
Boundary Conditions 261
Particle-Based Simulation 263
Implicit Solvent: Brownian Dynamics 264
Explicit Solvent: Molecular Dynamics 267
Flux-Based Simulation 273

Nernst–Planck Equation 274
The Poisson–Nernst–Planck (NP) Method 278
Hierarchical Simulation Schemes 282
Future Directions and Concluding Remarks 283
References 284

5. Wavelets in Chemistry and Chemoinformatics 295
C. Matthew Sundling, Nagamani Sukumar, Hongmei Zhang, Mark J. Embrechts, and Curt M. Breneman

Preface 295
Introduction to Wavelets 296
Fourier Transform 297
Continuous Fourier Transform 297
Short-Time Fourier Transformation 298
Wavelet Transform 300
Continuous Wavelet Transform 301
Discrete Wavelet Transform 303
Wavelet Packet Transform 307
Wavelets vs. Fourier Transforms: A Summary 308
Application of Wavelets in Chemistry 309
Smoothing and Denoising 309
Signal Feature Isolation 312
Signal Compression 313
Quantum Chemistry 314
Classification, Regression, and QSAR/QSPR 316
Summary 321
References 321

Author Index 331

Subject Index 349

Contributors

Shela Aboud, Department of Molecular Biophysics and Physiology, Rush University, 1750 West Harrison Street, Chicago, IL 60612 U.S.A. (Electronic mail: saboud@ece.wpi.edu)

Andrij Baumketner, Institute for Condensed Matter Physics, 1 Svientsisky Street, Lviv, Ukraine (Electronic mail: andrij@icmp.lviv.ua)

Curt Breneman, Department of Chemistry and Chemical Biology, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180 U.S.A. (Electronic mail: brenec@rpi.edu)

Robert Eisenberg, Department of Molecular Biophysics and Physiology, Rush University, 1750 West Harrison Street, Chicago, IL 60612 U.S.A. (Electronic mail: beisenbe@rush.edu)

Mark Embrechts, Department of Decision Science and Engineering Systems, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180 U.S.A. (Electronic Mail: embrem@rpi.edu)

Emilio Esposito, Department of Chemistry and Molecular Biology, North Dakota State University, Fargo, ND 58105 U.S.A. (Electronic mail: emilio.esposito@ndsu.nodak.edu)

Miriam Friedel, Department of Physics, University of California at Santa Barbara, Santa Barbara, CA 93106-9530 U.S.A. (Electronic mail: mfriedel@physics.ucsb.edu)

Patrice Koehl, Department of Computer Science and Genome Center, University of California at Davis, 1 Shields Avenue, Davis, CA 95616 U.S.A. (Electronic mail: koehl@cs.ucdavis.edu)

Jeffry Madura, Department of Chemistry and Biochemistry, Duquesne University, Pittsburgh, PA 15282-1530 U.S.A. (Electronic mail: madura@duq.edu)

Marco Saraniti, Department of Electrical and Computer Engineering, Illinois Institute of Technology, 3301 South Dearborn Street, Chicago, IL 60616-3793 U.S.A. (Electronic mail: saraniti@iit.edu)

Joan-Emma Shea, Department of Chemistry and Biochemistry, University of California at Santa Barbara, Santa Barbara, CA 93106 U.S.A. (Electronic mail: shea@chem.ucsb.edu)

Nagamani Sukmar, Department of Chemistry and Chemical Biology, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180 U.S.A. (Electronic mail: nagams@rpi.edu)

C. Matthew Sundling, Department of Chemistry and Chemical Biology, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180 U.S.A. (Electronic mail: sundlm@rpi.edu)

Dror Tobi, Department of Computational Biology, School of Medicine, University of Pittsburgh, Pittsburgh, PA 15213 U.S.A. (Electronic mail: drt6@pitt.edu)

Hongmei Zhang, Department of Chemistry and Chemical Biology, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180 U.S.A. (Electronic mail: zhangh4@rpi.edu)

Contributors to Previous Volumes

Volume 1 (1990)

David Feller and **Ernest R. Davidson,** Basis Sets for Ab Initio Molecular Orbital Calculations and Intermolecular Interactions.

James J. P. Stewart, Semiempirical Molecular Orbital Methods.

Clifford E. Dykstra, Joseph D. Augspurger, Bernard Kirtman, and **David J. Malik,** Properties of Molecules by Direct Calculation.

Ernest L. Plummer, The Application of Quantitative Design Strategies in Pesticide Design.

Peter C. Jurs, Chemometrics and Multivariate Analysis in Analytical Chemistry.

Yvonne C. Martin, Mark G. Bures, and **Peter Willett,** Searching Databases of Three-Dimensional Structures.

Paul G. Mezey, Molecular Surfaces.

Terry P. Lybrand, Computer Simulation of Biomolecular Systems Using Molecular Dynamics and Free Energy Perturbation Methods.

Donald B. Boyd, Aspects of Molecular Modeling.

Donald B. Boyd, Successes of Computer-Assisted Molecular Design.

Ernest R. Davidson, Perspectives on Ab Initio Calculations.

Volume 2 (1991)

Andrew R. Leach, A Survey of Methods for Searching the Conformational Space of Small and Medium-Sized Molecules.

John M. Troyer and **Fred E. Cohen,** Simplified Models for Understanding and Predicting Protein Structure.

J. Phillip Bowen and **Norman L. Allinger,** Molecular Mechanics: The Art and Science of Parameterization.

Uri Dinur and **Arnold T. Hagler,** New Approaches to Empirical Force Fields.

Steve Scheiner, Calculating the Properties of Hydrogen Bonds by Ab Initio Methods.

Donald E. Williams, Net Atomic Charge and Multipole Models for the Ab Initio Molecular Electric Potential.

Peter Politzer and **Jane S. Murray,** Molecular Electrostatic Potentials and Chemical Reactivity.

Michael C. Zerner, Semiempirical Molecular Orbital Methods.

Lowell H. Hall and **Lemont B. Kier,** The Molecular Connectivity Chi Indexes and Kappa Shape Indexes in Structure-Property Modeling.

I. B. Bersuker and **A. S. Dimoglo,** The Electron-Topological Approach to the QSAR Problem.

Donald B. Boyd, The Computational Chemistry Literature.

Volume 3 (1992)

Tamar Schlick, Optimization Methods in Computational Chemistry.

Harold A. Scheraga, Predicting Three-Dimensional Structures of Oligopeptides.

Andrew E. Torda and **Wilfred F. van Gunsteren,** Molecular Modeling Using NMR Data.

David F. V. Lewis, Computer-Assisted Methods in the Evaluation of Chemical Toxicity.

Volume 4 (1993)

Jerzy Cioslowski, Ab Initio Calculations on Large Molecules: Methodology and Applications.

Michael L. McKee and **Michael Page,** Computing Reaction Pathways on Molecular Potential Energy Surfaces.

Robert M. Whitnell and **Kent R. Wilson,** Computational Molecular Dynamics of Chemical Reactions in Solution.

Roger L. DeKock, Jeffry D. Madura, Frank Rioux, and **Joseph Casanova,** Computational Chemistry in the Undergraduate Curriculum.

Volume 5 (1994)

John D. Bolcer and **Robert B. Hermann,** The Development of Computational Chemistry in the United States.

Rodney J. Bartlett and **John F. Stanton,** Applications of Post-Hartree–Fock Methods: A Tutorial.

Steven M. Bachrach, Population Analysis and Electron Densities from Quantum Mechanics.

Jeffry D. Madura, Malcolm E. Davis, Michael K. Gilson, Rebecca C. Wade, Brock A. Luty, and **J. Andrew McCammon,** Biological Applications of Electrostatic Calculations and Brownian Dynamics Simulations.

K. V. Damodaran and **Kenneth M. Merz, Jr.,** Computer Simulation of Lipid Systems.

Jeffrey M. Blaney and **J. Scott Dixon,** Distance Geometry in Molecular Modeling.

Lisa M. Balbes, S. Wayne Mascarella, and **Donald B. Boyd,** A Perspective of Modern Methods in Computer-Aided Drug Design.

Volume 6 (1995)

Christopher J. Cramer and **Donald G. Truhlar,** Continuum Solvation Models: Classical and Quantum Mechanical Implementations.

Clark R. Landis, Daniel M. Root, and **Thomas Cleveland,** Molecular Mechanics Force Fields for Modeling Inorganic and Organometallic Compounds.

Vassilios Galiatsatos, Computational Methods for Modeling Polymers: An Introduction.

Rick A. Kendall, Robert J. Harrison, Rik J. Littlefield, and **Martyn F. Guest,** High Performance Computing in Computational Chemistry: Methods and Machines.

Donald B. Boyd, Molecular Modeling Software in Use: Publication Trends.

Eiji Ōsawa and **Kenny B. Lipkowitz,** Appendix: Published Force Field Parameters.

Volume 7 (1996)

Geoffrey M. Downs and **Peter Willett,** Similarity Searching in Databases of Chemical Structures.

Andrew C. Good and **Jonathan S. Mason,** Three-Dimensional Structure Database Searches.

Jiali Gao, Methods and Applications of Combined Quantum Mechanical and Molecular Mechanical Potentials.

Libero J. Bartolotti and **Ken Flurchick,** An Introduction to Density Functional Theory.

Alain St-Amant, Density Functional Methods in Biomolecular Modeling.

Danya Yang and **Arvi Rauk,** The A Priori Calculation of Vibrational Circular Dichroism Intensities.

Donald B. Boyd, Appendix: Compendium of Software for Molecular Modeling.

Volume 8 (1996)

Zdenek Slanina, Shyi-Long Lee, and **Chin-hui Yu,** Computations in Treating Fullerenes and Carbon Aggregates.

Gernot Frenking, Iris Antes, Marlis Böhme, Stefan Dapprich, Andreas W. Ehlers, Volker Jonas, Arndt Neuhaus, Michael Otto, Ralf Stegmann, Achim Veldkamp, and **Sergei F. Vyboishchikov,** Pseudopotential Calculations of Transition Metal Compounds: Scope and Limitations.

Thomas R. Cundari, Michael T. Benson, M. Leigh Lutz, and **Shaun O. Sommerer,** Effective Core Potential Approaches to the Chemistry of the Heavier Elements.

Jan Almlöf and **Odd Gropen,** Relativistic Effects in Chemistry.

Donald B. Chesnut, The Ab Initio Computation of Nuclear Magnetic Resonance Chemical Shielding.

Volume 9 (1996)

James R. Damewood, Jr., Peptide Mimetic Design with the Aid of Computational Chemistry.

T. P. Straatsma, Free Energy by Molecular Simulation.

Robert J. Woods, The Application of Molecular Modeling Techniques to the Determination of Oligosaccharide Solution Conformations.

Ingrid Pettersson and **Tommy Liljefors,** Molecular Mechanics Calculated Conformational Energies of Organic Molecules: A Comparison of Force Fields.

Gustavo A. Arteca, Molecular Shape Descriptors.

Volume 10 (1997)

Richard Judson, Genetic Algorithms and Their Use in Chemistry.

Eric C. Martin, David C. Spellmeyer, Roger E. Critchlow, Jr., and **Jeffrey M. Blaney,** Does Combinatorial Chemistry Obviate Computer-Aided Drug Design?

Robert Q. Topper, Visualizing Molecular Phase Space: Nonstatistical Effects in Reaction Dynamics.

Raima Larter and **Kenneth Showalter,** Computational Studies in Nonlinear Dynamics.

Stephen J. Smith and **Brian T. Sutcliffe,** The Development of Computational Chemistry in the United Kingdom.

Volume 11 (1997)

Mark A. Murcko, Recent Advances in Ligand Design Methods.

David E. Clark, Christopher W. Murray, and **Jin Li,** Current Issues in De Novo Molecular Design.

Tudor I. Oprea and **Chris L. Waller,** Theoretical and Practical Aspects of Three-Dimensional Quantitative Structure–Activity Relationships.

Giovanni Greco, Ettore Novellino, and **Yvonne Connolly Martin,** Approaches to Three-Dimensional Quantitative Structure–Activity Relationships.

Pierre-Alain Carrupt, Bernard Testa, and **Patrick Gaillard,** Computational Approaches to Lipophilicity: Methods and Applications.

Ganesan Ravishanker, Pascal Auffinger, David R. Langley, Bhyravabhotla Jayaram, Matthew A. Young, and **David L. Beveridge,** Treatment of Counterions in Computer Simulations of DNA.

Donald B. Boyd, Appendix: Compendium of Software and Internet Tools for Computational Chemistry.

Volume 12 (1998)

Hagai Meirovitch, Calculation of the Free Energy and the Entropy of Macromolecular Systems by Computer Simulation.

Ramzi Kutteh and **T. P. Straatsma,** Molecular Dynamics with General Holonomic Constraints and Application to Internal Coordinate Constraints.

John C. Shelley and **Daniel R. Bérard,** Computer Simulation of Water Physisorption at Metal–Water Interfaces.

Donald W. Brenner, Olga A. Shenderova, and **Denis A. Areshkin,** Quantum-Based Analytic Interatomic Forces and Materials Simulation.

Henry A. Kurtz and **Douglas S. Dudis,** Quantum Mechanical Methods for Predicting Nonlinear Optical Properties.

Chung F. Wong, Tom Thacher, and **Herschel Rabitz,** Sensitivity Analysis in Biomolecular Simulation.

Paul Verwer and **Frank J. J. Leusen,** Computer Simulation to Predict Possible Crystal Polymorphs.

Jean-Louis Rivail and **Bernard Maigret,** Computational Chemistry in France: A Historical Survey.

Volume 13 (1999)

Thomas Bally and **Weston Thatcher Borden,** Calculations on Open-Shell Molecules: A Beginner's Guide.

Neil R. Kestner and **Jaime E. Combariza,** Basis Set Superposition Errors: Theory and Practice.

James B. Anderson, Quantum Monte Carlo: Atoms, Molecules, Clusters, Liquids, and Solids.

Anders Wallqvist and **Raymond D. Mountain,** Molecular Models of Water: Derivation and Description.

James M. Briggs and **Jan Antosiewicz,** Simulation of pH-dependent Properties of Proteins Using Mesoscopic Models.

Harold E. Helson, Structure Diagram Generation.

Volume 14 (2000)

Michelle Miller Francl and **Lisa Emily Chirlian,** The Pluses and Minuses of Mapping Atomic Charges to Electrostatic Potentials.

T. Daniel Crawford and **Henry F. Schaefer III,** An Introduction to Coupled Cluster Theory for Computational Chemists.

Bastiaan van de Graaf, Swie Lan Njo, and **Konstantin S. Smirnov,** Introduction to Zeolite Modeling.

Sarah L. Price, Toward More Accurate Model Intermolecular Potentials for Organic Molecules.

Christopher J. Mundy, Sundaram Balasubramanian, Ken Bagchi, Mark E. Tuckerman, Glenn J. Martyna, and **Michael L. Klein,** Nonequilibrium Molecular Dynamics.

Donald B. Boyd and **Kenny B. Lipkowitz,** History of the Gordon Research Conferences on Computational Chemistry.

Mehran Jalaie and **Kenny B. Lipkowitz,** Appendix: Published Force Field Parameters for Molecular Mechanics, Molecular Dynamics, and Monte Carlo Simulations.

Volume 15 (2000)

F. Matthias Bickelhaupt and **Evert Jan Baerends,** Kohn-Sham Density Functional Theory: Predicting and Understanding Chemistry.

Michael A. Robb, Marco Garavelli, Massimo Olivucci, and **Fernando Bernardi,** A Computational Strategy for Organic Photochemistry.

Larry A. Curtiss, Paul C. Redfern, and **David J. Frurip,** Theoretical Methods for Computing Enthalpies of Formation of Gaseous Compounds.

Russell J. Boyd, The Development of Computational Chemistry in Canada.

Volume 16 (2000)

Richard A. Lewis, Stephen D. Pickett, and **David E. Clark,** Computer-Aided Molecular Diversity Analysis and Combinatorial Library Design.

Keith L. Peterson, Artificial Neural Networks and Their Use in Chemistry.

Jörg-Rüdiger Hill, Clive M. Freeman, and **Lalitha Subramanian,** Use of Force Fields in Materials Modeling.

M. Rami Reddy, Mark D. Erion, and **Atul Agarwal,** Free Energy Calculations: Use and Limitations in Predicting Ligand Binding Affinities.

Volume 17 (2001)

Ingo Muegge and **Matthias Rarey,** Small Molecule Docking and Scoring.

Lutz P. Ehrlich and **Rebecca C. Wade,** Protein-Protein Docking.

Christel M. Marian, Spin-Orbit Coupling in Molecules.

Lemont B. Kier, Chao-Kun Cheng, and **Paul G. Seybold,** Cellular Automata Models of Aqueous Solution Systems.

Kenny B. Lipkowitz and **Donald B. Boyd,** Appendix: Books Published on the Topics of Computational Chemistry.

Volume 18 (2002)

Geoff M. Downs and **John M. Barnard,** Clustering Methods and Their Uses in Computational Chemistry.

Hans-Joachim Böhm and **Martin Stahl,** The Use of Scoring Functions in Drug Discovery Applications.

Steven W. Rick and **Steven J. Stuart,** Potentials and Algorithms for Incorporating Polarizability in Computer Simulations.

Dmitry V. Matyushov and **Gregory A. Voth,** New Developments in the Theoretical Description of Charge-Transfer Reactions in Condensed Phases.

George R. Famini and **Leland Y. Wilson,** Linear Free Energy Relationships Using Quantum Mechanical Descriptors.

Sigrid D. Peyerimhoff, The Development of Computational Chemistry in Germany.

Donald B. Boyd and **Kenny B. Lipkowitz,** Appendix: Examination of the Employment Environment for Computational Chemistry.

Volume 19 (2003)

Robert Q. Topper, David L. Freeman, Denise Bergin, and **Keirnan R. LaMarche,** Computational Techniques and Strategies for Monte Carlo Thermodynamic Calculations, with Applications to Nanoclusters.

David E. Smith and **Anthony D. J. Haymet**, Computing Hydrophobicity.

Lipeng Sun and **William L. Hase**, Born-Oppenheimer Direct Dynamics Classical Trajectory Simulations.

Gene Lamm, The Poisson-Boltzmann Equation.

Volume 20 (2004)

Sason Shaik and **Philippe C. Hibert**, Valence Bond Theory: Its History, Fundamentals and Applications. A Primer.

Nikita Matsunaga and **Shiro Koseki**, Modeling of Spin Forbidden Reactions.

Stefan Grimme, Calculation of the Electronic Spectra of Large Molecules.

Raymond Kapral, Simulating Chemical Waves and Patterns.

Costel Sârbu and **Horia Pop,** Fuzzy Soft-Computing Methods and Their Applications in Chemistry.

Sean Ekins and **Peter Swaan,** Development of Computational Models for Enzymes, Transporters, Channels and Receptors Relevant to ADME/Tox.

Volume 21 (2005)

Roberto Dovesi, Bartolomeo Civalleri, Roberto Orlando, Carla Roetti, and **Victor R. Saunders,** Ab Initio Quantum Simulation in Solid State Chemistry.

Patrick Bultinck, Xavier Gironés, and **Ramon Carbó-Dorca,** Molecular Quantum Similarity: Theory and Applications.

Jean-Loup Faulon, Donald P. Visco, Jr., and **Diana Roe,** Enumerating Molecules.

David J. Livingstone and **David W. Salt,** Variable Selection—Spoilt for Choice?

Nathan A. Baker, Biomolecular Applications of Poisson–Boltzmann Methods.

Baltazar Aguda, Georghe Craciun, and **Rengul Cetin-Atalay,** Data Sources and Computational Approaches for Generating Models of Gene Regulatory Networks.

CHAPTER 1

Protein Structure Classification

Patrice Koehl

Department of Computer Science and Genome Center, University of California, Davis, California

INTRODUCTION

The molecular basis of life rests on the activity of large biological macromolecules, including nucleic acids (DNA and RNA), carbohydrates, lipids, and proteins. Although each plays an essential role in life, there is something special about proteins, as they are the lead performers of cellular functions. As a response, structural molecular biology has emerged as a new line of experimental research focused on revealing the structure of these bio-molecules. This branch of biology has recently experienced a major uplift through the development of high-throughput structural studies aimed at developing a comprehensive view of the protein structure universe. Although these studies are generating a wealth of information that are stored into protein structure databases, the key to their success lies in our ability to organize and analyze the information contained in those databases, and to integrate that information with other efforts aimed at solving the mysteries behind cell functions. In this survey, the first step behind any such organization scheme, namely the classification of protein structures, is described. The properties of protein structures, with special attention to their geometry, are reviewed. Computer methods for the automatic comparison and classification of these structures are then reviewed along with the existing classifications of protein structures and their applications in biology, with a special focus on computational biology. The chapter concludes the review with a discussion of the future of these classifications.

Reviews in Computational Chemistry, Volume 22
edited by Kenny B. Lipkowitz, Thomas R. Cundari, and Valerie J. Gillet

Classification and Biology

Classification is a broad term that simply means putting things into classes. Any organizational scheme is a classification: Objects can be sorted with respect to size, color, origin, and so on. Classification is one of the most basic activities in any discipline of science, because it is easier to think about a few groups that have something in common than it is to think about each individual of a whole population. Scientific classification in biology started with Aristotle, in the fourth century B.C. He divided all livings things into two groups: animal and plant. Animals were divided into two groups: those with blood and those without (at least no red blood), whereas plants were divided into three groups based on their shapes. Aristotle was the first in a long line of biologists who classified organisms in an arbitrary, although logical way, to convey scientific information. Among these biologists is the Swedish naturalist Carolus Linnaeus from the eighteenth century who set formal rules for a two-name system called the binomial system of nomenclature, which is still used today. With the publication of '*On the Origin of Species*' by Darwin, the purpose of classification changed. Darwin argued that classification should reflect the history of life. In other words, species should be related based on a shared history. *Systematic classifications* were introduced accordingly, the aims of which are to reveal the *phylogeny*, i.e., the hierarchical structure by which every life-form is related to every other life-form. The recent advances in genetics and biochemistry, the wealth of information coming from genome sequencing projects, and the tools of bio-informatics are playing an essential role in the development of these new classification schemes, by feeding to the classifiers and taxonomists more and more data on the evolutionary relationships between species. Note that the genetic information used for classification is not limited to the sequence of the genes, but it also takes into account the products of these genes, and their contributions to the mechanisms of life. Because function is related to shape, protein structure classification will thus play a significant role in our understanding of the organization of life. Paraphrasing Jacques Monod[1], in the protein lies the secret of life.

The Biomolecular Revolution

All living organisms can be described as arrangements of cells, the smallest self-sustainable units capable of carrying functions important for life. Cells can be divided into organelles, which are themselves assemblies of biomolecules. These bio-molecules are usually polymers composed of smaller subunits whose atomic structures are known from standard chemistry. There are many remarkable aspects to this hierarchy, one of them being that it is ubiquitous to all life forms, from unicellular organisms to complex multicellular species. Unraveling the secrets behind this hierarchy has become one of the major

challenges for scientists in the twentieth and now twenty-first centuries. Although early research from the physics and chemistry communities has provided significant insight into the nature of atoms and their arrangements in small chemical systems, the focus is now on understanding the structure and function of bio-molecules. These usually large molecules serve as storage for the genetic information (the nucleic acids) and as key actors of cellular functions (the proteins). Biochemistry, one field in which these bio-molecules are studied, is currently experiencing a major revolution. In hope of deciphering the rules that define cellular functions, large-scale experimental projects are now being performed as collaborative efforts involving many laboratories in many countries to provide maps of the genetic information of different organisms (the *genome projects*), to derive as much structural information as possible on the products of the corresponding genes (the *structural genomics projects*), and to relate these genes to the function of their products, which is usually deduced from their structure (the *functional genomics projects*). The success of these projects is completely changing the landscape of research in biology. As of October 2004, more than 220 whole genomes have been fully sequenced and published, which corresponds to a database of over a million gene sequences,[2] and more than a thousand other genomes are currently being sequenced. The need to store these data efficiently and to analyze their contents has led to the emergence of a collaborative effort between researchers in computer science and biology. This new discipline is referred to as bio-informatics. In parallel, the repository of bio-molecular structures[3,4] contains more than 27,600 entries of proteins and nucleic acids. The same need to organize and analyze the structural information contained in this database is leading to the emergence of another partnership between computer science and biology, namely the discipline of bio-geometry. The combined efforts of researchers in bio-informatics and bio-geometry are expected to provide a comprehensive picture of the protein sequence and structure spaces, and their connection to cellular functions. Note that the emergence of these two disciplines is often viewed as a consequence of a paradigm shift in molecular biology,[5] because the classic approach of hypothesis-driven research in biochemistry is being replaced with a data-driven discovery approach. In reality the two approaches coexist, and both benefit from these computer-based disciplines.

Outline

Given the introduction to classification in biology and an update on the progress of research in structural biology, we can now examine protein structure classification, the topic of this chapter. The next section describes proteins and surveys their different levels of organization, from their primary sequence to their quaternary structure in cells. The following section surveys automatic methods for comparing protein structures and their application to classification. Then the existing protein structure classifications are described, focusing on the Structural Classification of Proteins (SCOP)[6]; the Class, Architecture,

Topology, and Homologous (CATH) superfamilies classification[7]; and the domain classification based on the Distance ALIgnment (DALI) algorithm.[8] Finally, the tutorial concludes with a discussion of the future of protein structure classifications.

BASIC PRINCIPLES OF PROTEIN STRUCTURE

Although all bio-molecules play an important role in life, there is something special about proteins, which are the products of the information contained in the genes. A finding that has crystallized over the last few decades is that geometric reasoning plays a major role in our attempt to understand the activities of these molecules. In this section, the basic principles that govern the shapes of protein structures are briefly reviewed. More information on protein structures can be found in protein biochemistry textbooks, such as those of Schulz and Schirmer,[9] Cantor and Schimmel,[10] Branden and Tooze,[11] and Creighton.[12] The reader is also referred to the excellent review by Taylor et al.[13]

Visualization

The need for visualizing bio-molecules is based on our early understanding that their shape determines their function. Early crystallographers who studied proteins could not rely (as it is common nowadays) on computers and computer graphics programs for representation and analysis. They had developed a large array of finely crafted physical models that allowed them to represent these molecules. Those models, usually made out of painted wood, plastic, rubber, or metal, were designed to highlight different properties of the molecule under study. In space-filling models, such as those of Corey–Pauling–Koltun (CPK),[14,15] atoms are represented as spheres, whose radii are the atoms' van der Waals radii. They provide a volumetric representation of the bio-molecules and are useful to detect cavities and pockets that are potential active sites. In skeletal models, chemical bonds are represented by rods, whose junctions define the position of the atoms. Those models were used for example by Kendrew et al. in their studies of myoglobin.[16] Such models are useful to chemists because they help highlight the chemical reactivity of the bio-molecule under study and, consequently, its potential activity. With the introduction of computer graphics to structural biology, the principles of these models have been translated into software such that molecules as well as some of their properties can now be visualized on a computer display. Figure 1 shows examples of computer visualizations of myoglobin, including space-filling and skeletal representations. Many computer programs are now available that allow one to visualize bio-molecules. Cited here are MOLSCRIPT[17] and VMD,[18] which have generated most of the figures of this chapter.

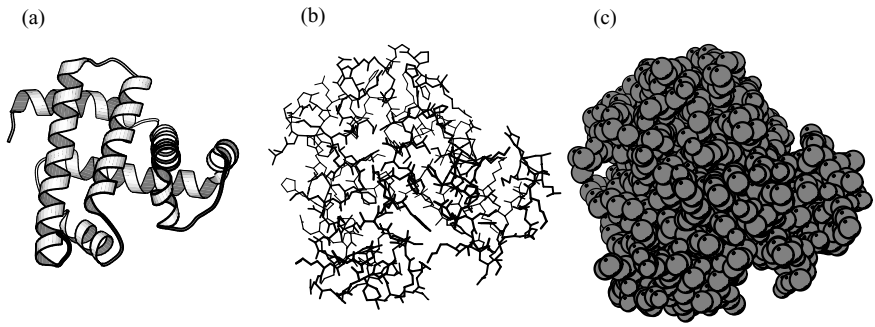


Figure 1 Visualizing protein structures. Myoglobin is a small protein very common in muscle cells, where it serves as oxygen storage. The structure of sperm whale myoglobin using three different types of visualization is depicted without the heme group. The coordinates are taken from the PDB file 1mbd. (a) **Cartoon.** This representation, also referred to as "ribbon" diagram, provides a high-level view of the local organization of the protein in secondary structures, shown as idealized helices. (b) **Skeletal model.** This representation uses lines to represent bonds; atoms are located at their endpoints where the lines meet. (c) **Space-filling diagram.** Atoms are represented as balls centered at the atoms, with radii equal to the van der Waals radii of the atoms. This representation shows the tight packing of the protein structure. Each of the representations is complementary to the others. Figure drawn using MOLSCRIPT.[17]

Protein Building Blocks

Proteins are heteropolymer chains of amino acids often referred to as *residues*. There are 20 naturally occurring amino acids that make up proteins. With the exception of proline, amino acids have a common structure, which is shown in Figure 2a. Naturally occurring amino acids that are incorporated into proteins are, for the most part, the levorotary (L) isomer. Substituents on the alpha carbon, called *side chains*, range in size from a single hydrogen atom to large aromatic rings. Those substituents can be charged, or they may include only nonpolar saturated hydrocarbons[19] (see Table 1 and Figure 2b). Nonpolar amino acids do not have a concentration of electric charges and are usually not soluble in water. Polar amino acids carry local concentration of charges and are either globally neutral, negatively charged (acidic), or positively charged (basic). Acidic and basic amino acids are classically referred to as electron acceptors and electron donors, respectively, which can associate to form salt bridges in proteins. Amino acids in solution are mainly dipolar ions: The amino group NH_2 accepts a proton to become NH_3^+, and the carboxyl group COOH donates a proton and becomes COO^-.

Protein Structure Hierarchy

Condensation between the -NH_3^+ and the -COO^- groups of two amino acids generates a peptide bond and results in the formation of a dipeptide.

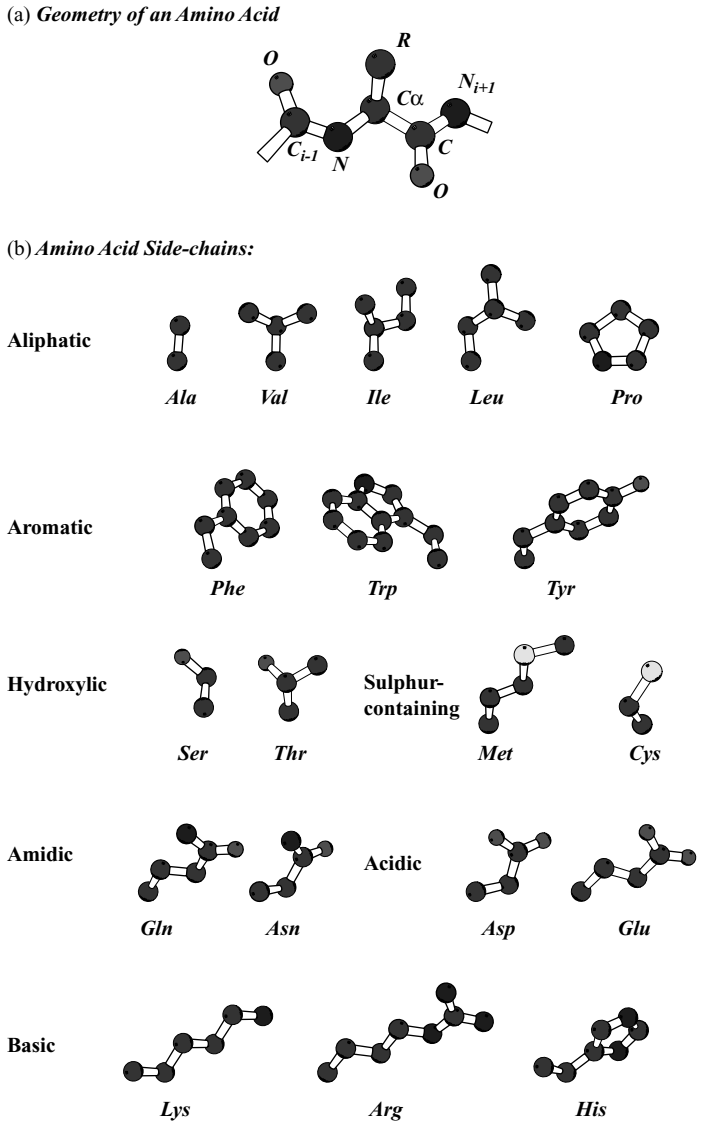


Figure 2 The twenty natural amino acids that make up proteins. (a) Each amino acid has a main-chain (N, C_α, C, and O) on which is attached a side-chain schematically represented as R. Amino acids in proteins are attached through planar peptide bonds, connecting atom C of the current residue to atom N of the following residue. For the sake of simplicity, the hydrogens are omitted. (b) Classification of the amino acid side-chains R is according to their chemical properties. Glycine (Gly) is omitted, as its side-chain is a single H atom.

Table 1 Classification of the 20 Amino Acids Based on Their Interaction With Water[19]

Classification	Amino Acid
Nonpolar	glycine (G)[a], alanine (A), valine (V), leucine (L), isoleucine (I), proline (P), methionine (M), phenylalanine (F), tryptophan (W)
Polar	serine (S), threonine (T), asparagine (N), glutamine (Q), cysteine (C), tyrosine (Y)
Acidic (polar)	aspartic acid (D), glutamic acid (E)
Basic (polar)	lysine (K), arginine (R), histidine (H)

[a] The one-letter code of each amino acid is given in parentheses.

Protein chains correspond to an extension of this chemistry, which results in long chains of many amino acids bonded together. The order in which amino acids appear defines the *sequence* or *primary structure* of the protein. In its native environment, the polypeptide chain adopts a unique three-dimensional shape, which is referred to as the *tertiary* or *native structure* of the protein.[20] The amino acid backbones are connected in sequence forming the protein *main-chain*, which frequently adopts canonical local shapes or *secondary structures*, mostly α-helices and β-strands (see Figure 3). α-helices form a right-handed helix with 3.6 amino acids per turn, whereas the β-strands form an approximately planar layout of the backbone. Helices often pack together to form a hydrophobic core, whereas β-strands pair together to form parallel or antiparallel β-sheets. In addition to these two types of secondary structures, a wide variety of other commonly occurring substructures, which are referred to as *super-secondary structures*. More information about these substructures can be found in the work of Efimov.[21–24]

Three Types of Proteins

Protein structures come in a large range of sizes and shapes. They can be divided into three major groups: *fibrous* proteins, *membrane* proteins, and *globular* proteins.

Fibrous proteins are elongated molecules in which the secondary structure is the dominant structure. Because they are insoluble in water, they play a structural or supportive role in the body and are involved in movement (such as in muscle and ciliary proteins). Fibrous proteins often (but not always) have regular repeating structures. Keratin, for example, which is found in hair and nails, is a helix of helices and has a seven-residue repeating structure. Silk, on the other hand, is composed only of β-sheets, with alternating layers of glycines and alanines and serines. In collagen, the major protein component of connective tissue, every third residue is a glycine and many others are prolines.

Membrane proteins are restricted to the phospholipid bilayer membrane that surrounds the cell and many of its organelles. These proteins cover a large

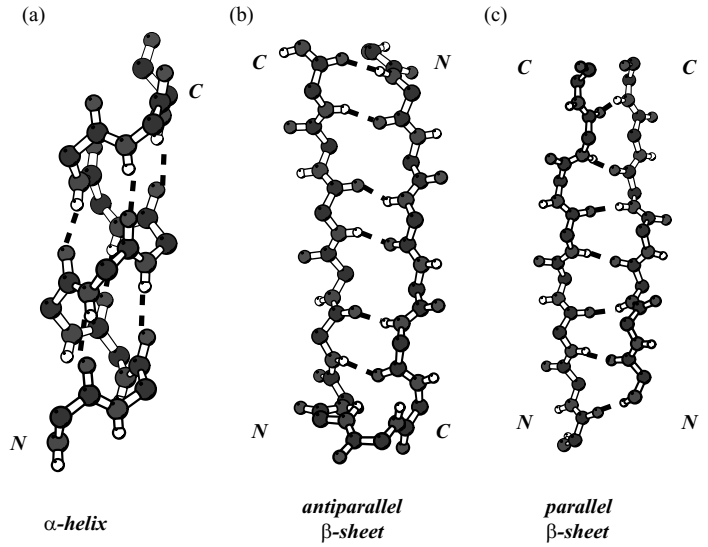
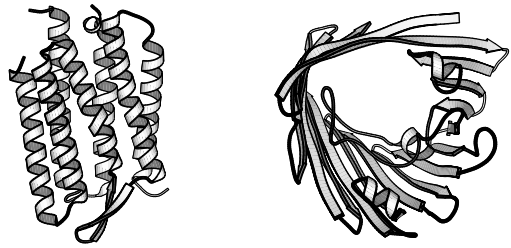


Figure 3 The three most common secondary structure elements (SSE) found in proteins. (a) The regular α-helix is a right-handed helix, in which all residues adopt similar conformations. The α-helix is characterized by hydrogen bonds between the oxygen O of residue *i*, and the polar backbone hydrogen HN (bound to N) of residue $i + 4$. Note that all C = O and N-HN bonds are parallel to the main axis of the helix. (b) An antiparallel β-sheet. Two strands (stretches of extended backbone segments) are running in an antiparallel geometry. The atoms HN and O of residue *i* in the first strand hydrogen bond with the atoms O and HN of residue *j* in the opposite strand, respectively, whereas residues $i + 1$ and $j + 1$ face outward. (c) A parallel β-sheet. The two strands are parallel, and the atoms HN and O of residue *i* in the first strand hydrogen bond with the O of residue *j* and the HN of residue $j + 2$, respectively. The same alternating pattern of residues involved in hydrogen bonds with the opposite strand, and facing outward is observed in parallel and antiparallel β-sheets. A strand can therefore be involved in two different sheets. For simplicity, side-chains and nonpolar hydrogens are ignored. The protein backbone is shown with balls and sticks, and hydrogen bonds are shown as discontinuous lines. Figure drawn using MOLSCRIPT.[17]

range of sizes and shapes, from globular proteins anchored in the membrane by means of a tail to proteins that are fully embedded in the membrane. Their function is usually to ensure transport of ions and small molecules like nutrients through the membrane. The structures of fully embedded membrane proteins can be placed into two major categories: the all helical structures, such as bacteriorhodopsin, and the all beta structures, such as porins (see Figure 4). As of October 2004, there are 158 structures of membrane proteins in the Protein Data Bank (PDB), out of which 86 are unique.

Globular proteins have a nonrepetitive sequence. They range in size from 100 to several hundred residues and adopt a unique compact structure. In globular proteins, nonpolar amino acid side chains have a tendency to cluster together to form the interior, hydrophobic core of the proteins, whereas the



(a) Bacteriorhodopsin (b) Porin

Figure 4 Two examples of membrane proteins. (a) Bacteriorhodopsin is mainly an α-protein containing seven helices. It is a membrane protein serving as an ion pump and is found in bacteria that can survive in high salt concentrations. (b) Porin is a β-barrel. Porins work as channels in cell membranes, which let small metabolites such as ions and amino acids in and out of the cell. Figure drawn using MOLSCRIPT.[17]

hydrophilic polar amino acid side chains remain accessible to the solvent on the exterior of the "glob." In the tertiary structure, β-strands are usually paired in parallel or antiparallel arrangements to form β-sheets. On average, a protein main-chain consists of about 25% of residues in α-helix formation and 25% of residues in β-strands, with the rest of the residues adopting less-regular structural arrangements.[25]

Geometry of Globular Proteins

From the seminal work of Anfinsen,[26] we know that the sequence fully determines the three-dimensional structure of a protein, which itself defines its function. Although the key to the decoding of information contained in genes was found more than 50 years ago (the genetic code), we have not yet rigorously defined the rules relating a protein sequence to its structure.[27,28] Ongoing work in the area of predicting protein structure based on sequence is the topic of Chapter 3 by Shea et al.[29] Our knowledge of protein structure comes from years of experimental studies, primarily using X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy. The first protein structures to be solved were those of myoglobin and hemoglobin.[16,30] There are now over 27,700 protein structures in the PDB database[3,4] (see http://www.rcsb.org). It is to be noted that this number overestimates the actual number of different structures available because the PDB is redundant; i.e., it contains several copies of the same proteins, with minor mutations in the sequence and no changes in the structure.

Because only two types of secondary structures (α and β) exist, proteins can be divided into three main structural classes.[31] These are mainly α proteins,[32] mainly β proteins,[33–35] and mixed α–β proteins.[36] A fourth class includes proteins with little or no secondary structures at all that are stabilized by metal ions and/or disulphide bridges. A significant effort has been made by

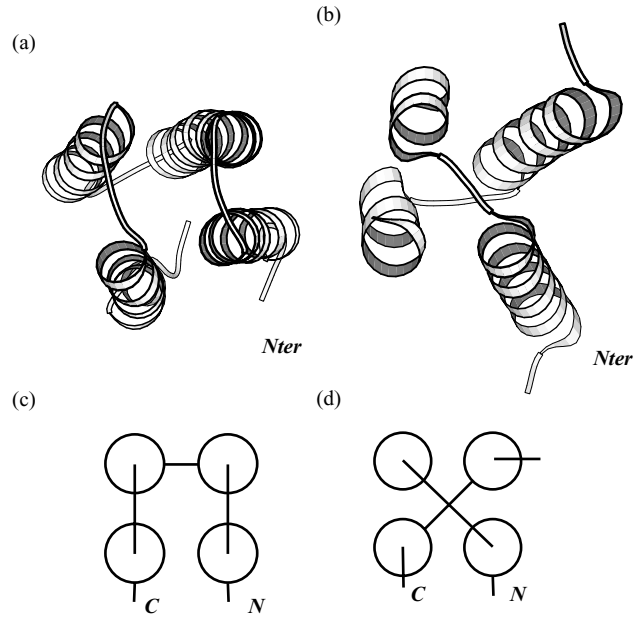


Figure 5 Two different topologies of four-helix bundles. A bundle is an array of α-helices, each oriented roughly along the same (bundle) axis. (a) and (c) show a four helical, up-and-down bundle with a left-handed twist, observed in hemerythrin from a sipunculid worm. (b) and (d) show a four helix bundle with a right handed twist, observed in a fragment of the dimerization domain of a liver transcription factor. (a) and (b) are cartoon representations of the proteins obtained with MOLSCRIPT,[17] whereas (c) and (d) show the schematic topologies produced by TOPS (http://www.tops.leed.ac.uk/).

scientists to a folding class to proteins automatically; these efforts will be reviewed in the next section. There has also been significant work on predicting a protein folding class based on its sequence, the details of which can be found in Refs. 37–44.

The α class, the smallest of the three major classes, is dominated by small proteins, many of which form a simple bundle of α helices packed together to form a hydrophobic core. A common motif in the mainly α class is the four helix bundle structure, which is depicted in Figure 5. The most extensively studied α structure is the globin fold, which has been found in a large group of related proteins, including myoglobin and hemoglobin. This structure includes eight helices that wrap around the core to form a pocket where a heme group is bound.[16]

The β class contains the parallel and antiparallel β structures. The β strands are usually arranged in two β sheets that pack against each other and form a distorted barrel structure. Three major types of β barrels exist, the up-and-down barrels, the Greek key barrels,[45] and the jelly roll barrels (see Figure 6). Most known antiparallel β structures, including the

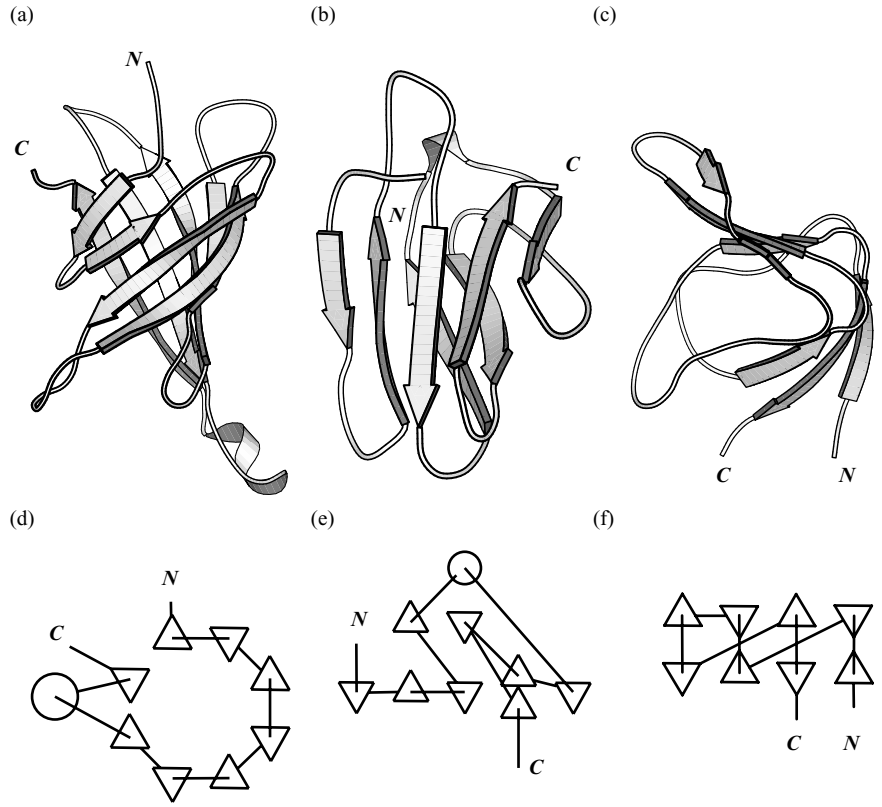


Figure 6 Three common sandwich topologies of beta proteins: a meander (a and d) observed in a glycoprotein from chicken, a Greek key (b and e) observed in and α-amylase (PDB code 1bli), and a jelly roll (c and f) observed in a gene activator protein from *E. Coli* (PDB code 1g6n). A meander (or up-and-down) is a simple topology in which any two consecutive strands are adjacent and antiparallel. A Greek key motif is a topology of a small number of β-sheet strands in which some inter-strand connection exist between β-sheets. The jelly roll topology is a variant of the Greek key topology with both ends crossed by two inter-strand connections. a, b, and c are cartoon representations of the proteins obtained with MOLSCRIPT,[17] while d, e and f show the schematic topologies produced by TOPS (http://www.tops.leed.ac.uk/).

immunoglobulins, have barrels that include at least one Greek key motif. The two other motifs are observed in proteins of diverse function, where functional diversity is obtained by differences in the loop regions connecting the β strands. β structures are often characterized by the number of β-sheets in the structure and the number and direction of the strands in the sheet. It leads to a rigid classification scheme,[46] which is sensitive to the definition of hydrogen bonds and β-strands.

The α–β protein class is the largest of the three classes. It is subdivided into proteins having an alternating arrangement of α helices and β strands

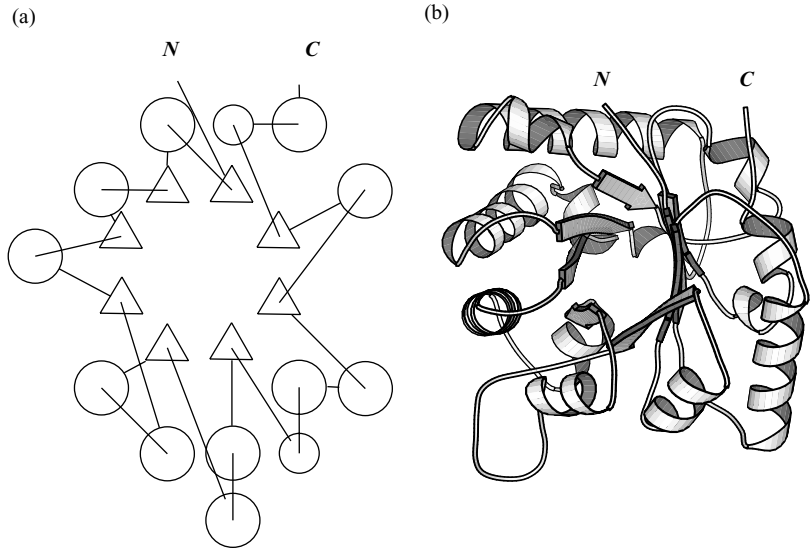


Figure 7 Topology (a) **and cartoon representation** (b) **of the TIM barrel**. The protein chain alternates between β and α secondary structure type, giving rise to a barrel β-sheet in the center surrounded by a large ring of α-helix on the outside. This structure, first seen in the triose phosphate isomerase of chicken, has been observed in many unrelated proteins since then. The topology is drawn using TOPS (http://www.tops.leed.ac.uk/), and the cartoon is generated using MOLSCRIPT.[17]

along the sequence and those with more segregated secondary structures. The former subclass is divided into two groups: one with a central core of (often eight) parallel β strands arranged as a barrel surrounded by α helices, and a second group consisting of an open, twisted parallel or mixed β sheet, with α helices on both sides (see Figure 7). A particularly striking example of a β–α barrel is seen in the eight-fold β–α barrel $(\beta\alpha)_8$ that was found originally in the triose phosphate isomerase of chicken,[47] and is often referred to as the TIM-barrel (for a complete analysis, see Refs. 48–55). Many proteins adopting a TIM barrel structure have completely different amino acid sequences and different biological functions. The open α/β-sheet structures vary considerably in size, number of β strands, and their strand order.

Protein Domains

Large proteins do not contain a single large hydrophobic core, probably because of limitations in their folding kinetics and stability. Large proteins are organized into "units" with sizes around 200–300 residues, which are referred to as *domains*.[56–58] Single compact units of more than 500 amino acids are rare. For a detailed analysis of domains in proteins, see Ref. 59. There are five different working definitions of protein domains: (1) regions that display

a significant level of sequence similarity; (2) the minimal part of a protein that is capable of performing a function; (3) a region of a protein with an experimentally assigned function; (4) a region of a structure that recurs in different contexts in different proteins; and (5) a compact, spatially distinct unit of protein structure. As more structures of proteins are solved, contradictions in these definitions appear. Some domains are compact, whereas others are clearly not globular. Some are too small to form a stable domain and thus lack a hydrophobic core. Currently, we are in the awkward situation in which the concept of a structural domain is well accepted, yet its definition is ambiguous;[60] this will be discussed in detail in the next section.

Resources on Protein Structures

Many resources related to protein structure and function exist; the Web addresses of these services are compiled in Table 2. Almost all experimental

Table 2 Resources on Protein Structures

Scheme	Description	Web Address
PDB	Repository of protein structures	http://www.rcsb.org/
PDB at a Glance	Interface to PDB	http://cmm.info.nih.gov/modeling/pdb_at_a_glance.html
Molecules to Go	Interactive interface to the PDB	http://molbio.info.nih.gov/cgi-bin/pdb/
MSD	EBI interface to the PDB, with integration to EBI resources	http://www.ebi.ac.uk/msd/
PDBSum	Summaries and structural analyses of PDB files	http://www.ebi.ac.uk/thornton-srv/databases/pdbsum
Biotech Validation Suite	Suite of programs that generates a quality control on protein structures	http://biotech.ebi.ac.uk:8400/
NRL_3D	Sequence-structure databases	http://laguerre.psc.edu/general/software/packages/nrl_3d/
Entrez	NCBI databases	http://www.ncbi.nlm.nih/gov/Database/index.html
SRS	Sequence Retrieval Services (includes structural information)	http://srs.embl-heidelberg.de:8000/srs5/
DSSP	Database of secondary structures of proteins (available through SRS)	http://srs.embl-heidelberg.de:8000/srs5/
TOPS	Generates a cartoon of the topology of a protein	http://www.tops.leeds.ac.uk/
PISCES	Protein sequence culling server: generates subsets of PDB based on users' criteria	http://dunbrack.fccc.edu/PISCES.php/
ASTRAL	Databases and tools for analyzing protein structure; derived from SCOP	http://astral.berkeley.edu/

protein structures publicly available today are stored in the PDB,[3] a database maintained through the RCSB consortium,[4] and available on the Web at http://www.rcsb.org/. Many services have been developed to supplement the PDB to ease access to the information it contains. For example, the services "PDB at a glance" and "Molecules to Go" were designed as easy-to-use interfaces with simple search engines. The MSD relational database is derived from the PDB and has the aim of providing a knowledge discovery and data mining environment for biological structure data. PDBSum[61,62] and the Biotech Validation Suite are services that allow a user to check the quality of a protein structure. NRL, Entrez, and SRS are integrated services that regroup the PDB with other databases containing information about proteins. For example, SRS includes DSSP,[63] a database of secondary structures of proteins. PISCES[64] and ASTRAL[65–67] can generate subsets of the PDB database, based on the user's criteria.

PROTEIN STRUCTURE COMPARISON

Any attempts to study a large collection of objects usually start with classifying them according to a given measure of similarity. Protein structure similarity is most often detected and quantified by a protein structure alignment program, which is applied to the different domains of the proteins considered. In this section, existing techniques for automatically detecting domains in protein structures are reviewed along with techniques for finding the optimal alignment between two structural domains. The section concludes with a brief description of new techniques for comparing protein structural domains that do not rely on a structural alignment, but instead rely on a direct comparison of the topology of the domains.

Automatic Identification of Protein Structural Domain

Decomposition of multidomain protein structures into individual domains has traditionally been done manually. Because the rate of protein structure determination has increased drastically in the past few years, this manual process has now become a bottleneck in maintaining and updating protein structure classifications; there is a need for automation. Automatic decomposition of proteins into structural domains can be traced back to the work of Rossman and Liljas in 1974,[68] who used $C\alpha$-$C\alpha$ distance maps. They suggested that a domain has, internally, many short residue–residue distances, but few short distances with the rest of the protein. Their analysis of the distance plots, however, required human intervention. Crippen[20] generalized this concept using hierarchical cluster analysis to locate protein fragment–fragment contacts. His procedure generates a tree of protein fragments, from a small, locally compact region of the complete protein. Several methods have

been proposed subsequently that follows this concept of identifying domains based on a difference between intradomain and interdomain properties. Some of these methods follow up on the work of Rossman and Liljas and compare intradomain and interdomain distances,[69–72] whereas others evaluate contact surface area between domains,[73,74] the "compactness" of the domains,[56,75,76] or their dynamics.[77] Recursive algorithms have been developed to find the cutting points that delineate domains in a protein chain. These algorithms either scan the chain to find single cuts such that the two resulting fragments verify a given protein domain definition based on one of the properties enumerated above or look directly for multiple cuts (see, for example, Ref. 71). The problem of delineating protein domains has also been formulated as an eigenvalue problem on the Cα-Cα distance matrix,[77] as well as a network flow problem.[78,79]

These methods take the approach in which a predefined domain definition is imposed on the structural data. In the language of systems analysis, such methods are referred to as "top-down" approaches, and the inherent problem in their applications is the difficulty in recognizing when the data fit or do not fit the model. An alternative approach is to reverse the direction and let the model emerge from the data, in what is often referred to as a "bottom-up" approach. Taylor[80] recently developed a "bottom-up" approach to identify domains in protein, using an Ising model, in which the structural elements of the model change state according to a function of the state of the neighbors. His procedure works as follows. Each residue in the protein chain is assigned a numeric label, usually the sequential residue number. If a residue i with label s_i is surrounded by neighbors with, on average, a higher label, its label increases; otherwise it decreases. This procedure is iterated until the system reaches equilibrium. Special care is taken to ensure (1) that the protein chain does not pass between domains too frequently; (2) that secondary structures, in particular, β-sheets are not broken; and (3) that small domains are either ignored or avoided. Swindells developed an alternative "bottom-up" approach, in which he first identifies core regions in the protein,[81] which are then extended to define the different domains in the proteins.[82]

Most existing methods for identifying protein domains include a refinement scheme to assess the quality of the domains that have been identified. Domain quality is computed according to accessible surface area, hydrophobic moment profile, size, compactness, number of protein segments involved,[79] and presence of intact β sheets.[80]

The diversity of definitions for protein structural domains is a serious issue for the generation of protein structure classifications. Many programs have been developed to delineate domains automatically in multidomain proteins. Table 3 lists the programs that are currently accessible on the Web, either as a Web service or for download. Although the results of these programs agree in most cases, discrepancies still prevent consistent assignments of protein domains.[60] The absence of quality control in the results of protein domain assignment programs has led researchers to use a combination of

Table 3 Websites for Publicly Available Services or Programs for Protein Domain Assignment

Program	Web Access
DIAL	http://www.ncbs.res.in/~faculty/mini/ddbase/dial.html
DomainParser	http://compbio.ornl.gov/structure/domainparser
DOMAK	http://www.compbio.dundee.ac.uk/Software/Domak/domak.html
PDP	http://123d.ncifcrf.gov/pdp.html

automatic and manual methods. For example, CATH[7] defines domains in multidomain proteins based on a consensus of three automatic programs, namely PUU,[77] DOMAK,[83] and Detective.[82] When all three programs agree on an assignment, the corresponding domains are included in CATH. In cases of disagreement, the domains are assigned manually, either from visual inspection or from information available in the literature or on the Web. Several structural domain databases are available on the Web to assist manual assignments of domains (see Table 4).

The Rigid-Body Transformation Problem

Definition

Before one attempts to classify protein structures, it is important to evaluate structure similarities. Many ways exist in which protein structures can be compared, that will be reviewed below. Most of these approaches proceed in two steps: (1) find the transformation that provides the optimal superposition between the two structures, and (2) define the similarity score as the distance between the two structures after superposition. This section describes how to obtain the optimal transformation for step (1).

We begin with the (relatively) easy problem of comparing two protein structures with the same number of atoms and a known correspondence table between these atoms (for a review, see Ref. 84). This problem is often solved when comparing two possible models for the structure of a protein. Because it is such a common problem, and because there still exists some confusion about

Table 4 Databases of Protein Structural Domains

Database	Web Access	Method
3Dee	http://www.compbio.dundee.ac.uk/3Dee	DOMAK
Authors	http://www.bmm.icnet.uk/~domains/test/dom-rr.html	Domains identified in the literature
DALI	http://www.ebi.ac.uk/dali/domain/3.1beta	DALI Domain Definition
DDBASE	http://www.ncbs.res.in/~faculty/mini/ddbase/ddbase.html	DIAL

how it can be solved,[85] a full mathematical description of the problem, as well as a proof for one of its closed form solutions is given.

The problem of comparing two different models of a protein can be formalized as: given two sets of points $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots b_m)$ in three dimensional space and assuming that they have the same cardinality, i.e., n = m, and that the element a_i corresponds to the element b_i, find the optimal rigid body transformation G_{opt} between the two sets that minimizes a given distance metric D over all possible rigid body transformation G, as in Eq. [1]:

$$\min_{G}\{D(A - G(B))\} \quad [1]$$

When comparing two proteins, the sets of points can include the C_α only, all backbone atoms, or all atoms of the proteins. Different metrics have been used in the literature to determine the geometric similarity between sets of points. For protein superposition, the most common metric is the coordinate root mean square deviation (cRMS), which is defined as follows:

$$D(A, B) = cRMS(A, B) = \|A - B\| = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - b_i)^2} \quad [2]$$

A rigid body transformation is a transformation that does not produce changes in the size, shape, or topology of an object. Mathematically, it can be defined as a mapping $G: \Re^3 \rightarrow \Re^3$ that satisfies the properties:

$$\|G(x) - G(y)\| = \|x - y\| \text{ for all points } x \text{ and } y \quad [3]$$

and

$$G(x \wedge y) = G(x) \wedge G(y) \text{ for all vectors } x \text{ and } y \quad [4]$$

where $\wedge$ is the cross-product.

Equation [3] states that distances are conserved, whereas Eq. [4] says that internal reflection is not allowed. Rotations and translations are two examples of rigid body transformation, and in fact, a general rigid body transformation can be expressed as a combination of a rotation R and a translation T. The transformation problem can then be restated as finding the optimal rotation R and optimal translation T such that $\|A - RB - T\|$ is a minimum.

A Closed-Form Solution Based on Singular Value Decomposition

Many algorithms exist in the literature that solve the rigid transposition problem, coming from various fields including computer vision and image processing, robotics, astronomy, and computational biology. Those algorithms

differ with respect to the representation of the transformation and the minimization procedure. Some algorithms are based on closed-form solutions, whereas others use iterative solutions. For detailed descriptions of these algorithms, including comparison of their performances, the reader is referred to the surveys of Sabata and Aggarwal,[86] Ferrari and Guerra,[87] and Eggert et al.[88] Here a focus is placed on the representation typically used in computational biology. It is based on the singular-value decomposition (SVD)[89] of a correlation matrix C between the two sets of points.[90–93] This method seems to have been first derived by Schonenman in the context of factor analysis.[94] Other approaches include solutions based on a power decomposition of C[95] or on a representation of rotations with quaternions.[96–98] These methods have been shown to be equivalent.[88,98]

Using the definition of the metric given in Eq. [2], the rigid transformation problem can be restated as finding the rotation R_{min} and the translation T_{min} such that

$$\varepsilon = \frac{1}{n}\sum_{i=1}^{n}(a_i - Rb_i - T)^2 \qquad [5]$$

is minimum.

Considering variations with respect to T first, we find that for an extremum of ε,

$$\frac{\partial \varepsilon}{\partial T} = -\frac{2}{n}\sum_{i=1}^{n}(a_i - Rb_i - T) = 0 \qquad [6]$$

so that

$$T_{min} = \frac{1}{n}\sum_{i=1}^{n} a_i - R_{min}\left(\frac{1}{n}\sum_{i=1}^{n} b_i\right) = \mu_A - R_{min}\mu_B \qquad [7]$$

where μ_A and μ_B are the centers of mass of A and B, respectively.

Note that if the two sets of points are shifted such that their centers of mass coincide at the origin, $T_{min} = 0$. Let $x_i = a_i - \mu_A$ and $y_i = b_i - \mu_B$ be the coordinates of the shifted points, and let $X = [x_1, x_2, \ldots, x_n]$ and $Y = [y_1, y_2, \ldots y_n]$ be the $3 \times n$ matrices representing the two sets of points A and B, after shifting. The rigid-body transformation problem can then be restated as finding the optimal rotation matrix R_{min} such that

$$\varepsilon = \frac{1}{n}\| X - RY \|^2 \qquad [8]$$

is minimum.

Let C be the correlation matrix of X and Y:

$$C = XY^T \rightarrow C_{ij} = \sum_{k=1}^{n} x_{ik} y_{jk}, i, j = 1, 2, 3 \quad [9]$$

and UDV^T be an SVD[89] of C ($UU^T = VV^T = I$, $D = \mathrm{diag}(d_i)$, $d_1 \geq d_2 \geq d_3 \geq 0$). The minimum value of ε with respect to R is then

$$\varepsilon_{min} = \frac{1}{n}\left(\|X\|^2 + \|Y\|^2 - 2(d_1 + d_2 + \lambda d_3)\right) \quad [10]$$

where $\lambda = \mathrm{sign}(\det(C))$. The optimal rotation is given by

$$R_{\min} = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \lambda \end{pmatrix} V^T \quad [11]$$

when $\mathrm{rank}(C) \geq 2$.

This result was first formulated by Schonemann,[94] later refined by Arun et al.,[93] Horn et al.,[95] and Umeyama.[99] Here the proof of Umeyama is given. Finding a rotation matrix R that minimizes ε can be rewritten as finding a matrix R that minimizes the objective function O defined as

$$O = \|X - RY\|^2 + tr\left(L\left(R^T R - I\right)\right) + g(\det(R) - 1) \quad [12]$$

where g is a Lagrange multiplier and L is a symmetric matrix of Lagrange multipliers. The second and third term of O represent the conditions for R to be an orthogonal and proper rotation matrix, respectively. Partial differentiations of O with respect to R, L, and g lead to the following system of equations:[99]

$$\frac{\partial O}{\partial R} = -2XY^T + 2RYY^T + 2RL + gR = 0 \quad [13]$$

$$\frac{\partial O}{\partial L} = R^T R - I = 0 \quad [14]$$

$$\frac{\partial O}{\partial g} = \det(R) - 1 = 0 \quad [15]$$

From Eq. [13],

$$RM = XY^T = C \quad [16]$$

where C is the covariance matrix defined in Eq. [9] and M is a symmetric 3×3 matrix defined by

$$M = YY^T + L + \frac{g}{2}I \qquad [17]$$

Transposing Eq. [16], we obtain

$$MR^T = C^T \qquad [18]$$

and multiplying each side of [16] with each side of [18], Eq. [19] is obtained, as $R^TR = I$ (Eq. [14]):

$$M^2 = C^TC = VD^2V^T \qquad [19]$$

Because M and M^2 are commutative ($MM^2 = M^2M$), both can be reduced to diagonal form by the same orthogonal matrix. Thus,

$$M = VDSV^T \qquad [20]$$

where $S = \text{diag}(s_i)$, $s_i = 1$ or -1.

From Eq. [20],

$$\det(M) = \det\left(VDSV^T\right) = \det(D)\det(S) \qquad [21]$$

and from Eq. [16]

$$\det(M) = \det(R^T)\det(C) = \det(C) \qquad [22]$$

as $\det(R) = \det(R^T) = 1$ (Eq. [15]).

Thus,

$$\det(D)\det(S) = \det(C) \qquad [23]$$

Singular values are non-negative, $\det(D) = d_1d_2d_3 \geq 0$; hence, $\det(S)$ must be equal to 1 if $\det(C) > 0$ and -1 if $\det(C) < 0$.

From the properties of norm and trace of a matrix, we get

$$\begin{aligned}\varepsilon &= \frac{1}{n}tr\left((X - RY)(X - RY)^T\right) = \frac{1}{n}\left(tr\left(XX^T\right) + tr\left((RY)(RY)^T\right) - 2tr\left(XY^TR^T\right)\right) \\ &= \frac{1}{n}\left(\|X\|^2 + \|RY\|^2 - 2tr\left(XY^TR^T\right)\right) = \frac{1}{n}\left(\|X\|^2 + \|Y\|^2 - 2tr(M)\right)\end{aligned} \qquad [24]$$

Substituting Eq. [20] into Eq. [24], we have

$$\begin{aligned}\varepsilon &= \frac{1}{n}\left(\|X\|^2 + \|Y\|^2 - 2tr\left(VDSV^T\right)\right) \\ &= \frac{1}{n}\left(\|X\|^2 + \|Y\|^2 - 2tr(DS)\right) \\ &= \frac{1}{n}\left(\|X\|^2 + \|Y\|^2 - 2(d_1s_1 + d_2s_2 + d_3s_3)\right)\end{aligned} \quad [25]$$

Thus, the minimum value of ε is achieved when $s_1 = s_2 = s_3 = 1$ if $\det(C) > 0$, and $s_1 = s_2 = 1$, $s_3 = -1$ if $\det(C) < 0$.

Next, we determine a rotation matrix R achieving the above minimum value. When $\text{rank}(C) = 3$, M is nonsingular and its inverse is given by

$$M^{-1} = (VDSV^T)^{-1} = VSD^{-1}V^T = VD^{-1}SV^T \quad [26]$$

and

$$R_{min} = CM^{-1} = UDV^TVD^{-1}SV^T = USV^T \quad [27]$$

which completes the proof for Eq. [10]. Note that this expression for R_{min} is also valid when $\text{rank}(C) = 2$ (see Ref. 99).

Weighted Superposition of Sets of Points

It is not always proper to assign the same importance to all points of A and B. Thus, variants of the rigid-body transformation problem have been developed in which each point i is given a weight ω_i. Examples of weighting schemes include (1) considering the mass of the atoms included in the superposition, (2) giving different weights to atoms of the backbone of the protein compared with atoms of the side chains, and (3) giving greater weights to atoms belonging to secondary structures of the protein. Solving the weighted variant of the rigid-body transformation problem amounts to finding the optimal translation T and optimal rotation R such that Eq. [28] is a minimum.

$$\varepsilon' = \frac{1}{n}\sum_{i=1}^{n}\omega_i(a_i - Rb_i - T)^2 \quad [28]$$

Considering variations with respect to T first, we find that for an extremum of ε',

$$\frac{\partial\varepsilon'}{\partial T} = -\frac{2}{n}\sum_{i=1}^{n}\omega_i(a_i - Rb_i - T) = 0 \quad [29]$$

so that

$$T_{min} = \frac{1}{\Omega}\sum_{i=1}^{n} \omega_i a_i - R_{min}\left(\frac{1}{\Omega}\sum_{i=1}^{n} \omega_i b_i\right) = \mu'_A - R_{min}\mu'_B \qquad [30]$$

where Ω is the sum of the weights ($\Omega = \sum_{i=1}^{n} \omega_i$), and μ'_A and μ'_B are the weighted centers of mass of A and B, respectively.

Note again that if the two sets of points are shifted such that their weighted barycenters coincide at the origin, $T_{min} = 0$. Let $x'_i = \sqrt{\omega_i}(a_i - \mu'_A)$ and $y'_i = \sqrt{\omega_i}(b_i - \mu'_B)$ be the weighted coordinates of the shifted points, and let $X' = [x'_1, x'_2, \ldots, x'_n]$ and $Y' = [y'_1, y'_2, \ldots y'_n]$ be the $3 \times n$ matrices representing the two weighted sets of points A and B, after shifting. The rigid-body transformation problem can then be restated as finding the optimal rotation matrix R_{min} such that

$$\varepsilon' = \frac{1}{n}\| X' - RY'\|^2 \qquad [31]$$

is minimum.

Equation [31] is equivalent to Eq. [8], and the same algorithm is used to solve it.

A General Algorithm for Point Set Superposition

The general procedure for superposing two protein structures, when the equivalent atoms are known, can then be summarized as follows:

1. Set input points $A = (a_1, a_2, \ldots a_n)$ for protein 1, $B = (b_1, b_2, \ldots, b_n)$ for protein 2, and weights $(\omega_1, \omega_2, \ldots, \omega_n)$.
2. Compute weighted centers of mass of A and B:

$$\mu'_A = \frac{\sum_{i=1}^{n} \omega_i a_i}{\sum_{i=1}^{n} \omega_i}; \quad \mu'_B = \frac{\sum_{i=1}^{n} \omega_i b_i}{\sum_{i=1}^{n} \omega_i}$$

3. Generate the weighted covariance matrix:

$$C'_{ij} = \sum_{k=1}^{n} \omega_k (a_{ki} - \mu'_{Ai})(b_{kj} - \mu'_{Aj}), \quad i = 1,2,3; \quad j = 1,2,3$$

4. Compute the SVD of C': $C' = UDV^T$ and $\lambda = \text{sign}(\det(C'))$, noting that $D = \text{diag}(d_1,d_2,d_3)$ with $d_1 \geq d_2 \geq d_3 \geq 0$.

5. Define optimal rotation $R_{min} = USV^T$, with $S = \text{diag}(1,1,\lambda)$, and optimal translation

$$T_{min} = \mu'_A - R_{min}\ \mu'_B$$

6. Compute the cRMS between the two structures:

$$cRMS = \sqrt{\varepsilon'} = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{n} \omega_i (a_i - \mu'_A)^2 + \sum_{i=1}^{n} \omega_i (b_i - \mu'_B)^2 - 2(d_1 + d_2 + \lambda d_3)\right)}$$

This algorithm does not take into account the possibility of "noise" in the coordinates of the points. For proteins, the coordinates of atoms are approximations to a "true" position: Proteins are flexible, with fluctuation about a mean position. Moreover, the physical experiments that provide information on the coordinates (usually X-ray crystallography and NMR spectroscopy) have a degree of experimental uncertainty. When superposing two models for the structure of one protein, the cRMS value is therefore a combination of the actual fluctuation between the two models and of the noise level contained in the two models. Noise is even more important for the superposition of two proteins of different lengths.

Protein Structure Superposition

An Ambiguous Problem

The problem of finding an optimal alignment between two proteins is more complex than just solving the rigid-body transformation problem, because the correspondence, i.e., the list of equivalent residues in the two proteins, is often not known. Indeed, the correspondence is part of the desired output, along with the optimal transformation of the position of one protein with respect to the other. The protein structure alignment problem can be stated as finding the maximal substructures of the two proteins that exhibit the highest degree of similarity.

A "substructure" of protein A is a subset of its points, arranged by order of appearance in A. We denote the substructure defined by $P = (p_1, p_2, \ldots, p_k)$, where $1 \leq p_1 < p_2 \ldots < p_k \leq n$, by $A(P) = (a_{p1}, a_{p2}, \ldots, a_{pk})$. The length $|A(P)|$ of A(P) is the number of points it contains, which in this case is k. A "gap" in A(P) is two consecutive indices p_i, p_{i+1} such that $p_i + 1 < p_{i+1}$.

Given two sets of points $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots b_m)$ in three-dimensional (3-D) space, the protein superposition problem is to find the optimal subsets $A(P)$ and $B(Q)$ with $|A(P)| = |B(Q)|$, and the optimal rigid-body transformation G_{opt} between the two subsets $A(P)$ and $B(Q)$ that

minimizes a given distance metric D over all possible rigid-body transformation G, i.e.,

$$\min_G\{D(A(P) - G(B(Q)))\} \quad [32]$$

The two subsets $A(P)$ and $B(Q)$ define a "correspondence," and $p = |A(P)| = |B(Q)|$ is called the correspondence length. Once the optimal correspondence is defined, it is easy to find the optimal rotation and translation using the rigid-body transformation algorithm described earlier. The concept of "optimal correspondence," however, requires more explanation. It is clear that $p = 1$ defines a trivial solution to the protein superposition problem: Any point of A can be aligned with any point of B, with a cRMS of 0. In practice, we are interested in finding the largest possible value for p under the condition that $A(P)$ and $B(Q)$ remain "similar."

A fast, reliable, and convergent method for protein structural alignment is not yet available although significant progress toward this goal has been made over the past decade.[100] Recent developments have focused on both the search algorithm and the definition of the target function to be minimized, which in turn provides a quantitative measure of the "similarity" between two structures. The most direct approach for comparing two protein structures is to move the set of points representing one structure as a rigid body over the set of points of the other structure and look for equivalent residues. It can be achieved only for relatively similar structures and will fail to detect local similarities of structures sharing common substructures. To avoid this problem, the structures can be decomposed into fragments [usually secondary structure elements (SSEs)], but this can lead to situations in which the global alignment is missed. Accordingly, recent work has focused on combining the local and global criteria in a hierarchical and heuristic approach. These methods proceed by first defining a list of equivalent positions in the two structures, from which a structural alignment can be derived. This initial equivalence set is defined by various methods, including dynamic programming,[101,102] comparing distance matrices,[8,103–105] fragment matching,[106,107] geometric hashing,[108–113] maximal common subgraph detection,[114–116] and local geometry matching.[117] Optimization of this equivalence set has been performed with dynamic programming,[102,118–120] Monte Carlo algorithms or simulated annealing,[8] a genetic algorithm,[121] incremental combinatorial extension of the optimal path,[122,123] and mean-field approaches.[124,125] Excellent reviews of these and other methods can be found in Refs. 13, 100, 126, and 127. Many groups involved in developing algorithms for protein structure alignment have generously made their programs available for use over the Internet and the World Wide Web. In some cases, the program is accessible for download, either as an executable or as a full source package (Table 5). These are

Table 5 Websites for Publicly Available Protein Structure Alignment Services and Programs

Program	Web Access (Interface)	Web Access (Program Download)	Method
CE	http://cl.sdsc.edu	ftp://ftp.sdsc.edu/pub/sdsc/biology/CE/src	Extension of the optimal path
DALILIGHT	http://www2.ebi.ac.uk/dali	http://ekhidna.biocenter.helsinki.fi:8080/dali/DaliLite/index.html	Distance matrix alignment
DEJAVU	http://portray.bmc.uu.se/cgi-bin/dejavu/scripts/dejavu.pl		Compare SSE[a]
FATCAT	http://fatcat.burnham.org/fatcatpair.html		Flexible structure alignment based on fragments
FoldMiner	http://dlb4.stanford.edu/FoldMiner/		Structure-database comparison based on motif search
K2 and K2SA	http://zlab.bu.edu/k2		Genetic algorithm (K2) or Simulated annealing (K2SA)
LOCK2	http://motif.stanford.edu/lock2/		Hierarchical protein structure superposition
LSQRMS	http://www.molmovdb.org/align/		STRUCTAL-based program
MATRAS	http://biunit.aist-nara.ac.jp/matras/		Markov transition model of evolution
PRIDE	http://hydra.icgeb.trieste.it/pride/		Probabilistic approach based on CA-CA distance matrix
PRISM	None	http://honiglab.cpmc.columbia.edu/	SSE alignment followed by iterative refinement of the equivalence list
PROSUP	http://lore.came.sbg.ac.at:8080/CAME/CAME_EXTERN/PROSUP		Hierarchical alignment
SARF2	http://123d.ncifcrf.gov/sarf2.html	http://123d.ncifcrf.gov/sarf2.html	Alignment of backbone fragments

Table 5 (*Continued*)

Program	Web Access (Interface)	Web Access (Program Download)	Method
SHEBA	http://rex.nci.nih.gov/RESEARCH/basic/lmb/mms/sheba.html	http://rex.nci.nih.gov/RESEARCH/basic/lmb/mms/SHEBA-download.html	Hierarchical alignment including profiles
SSAP	http://www.biochem.ucl.ac.uk/cgi-bin/cath/GetSsapRasmol.pl		Double dynamic program
SSM	http://www.ebi.ac.uk/msd-srv/ssm/	http://www.ebi.ac.uk/msd-srv/ssm/cgi-bin/ssmdcenter	Secondary Structure Matching
TOPS	http://balabio.dcs.gla.ac.uk/tops/versus.html	http://www.tops.leeds.ac.uk/	Alignment of simplified representations of proteins
TOPSCAN	http://www.bioinf.org.uk/topscan		Fast alignment based on SSE matching
VAST	http://www.ncbi.nlm.nih.gov/Structure/VAST/vastsearch.html		Vector alignment

[a]SSE: secondary structure elements

wonderful tools, and the reader is encouraged to test several of these sites. Many of these services have been tested on large datasets with known similarities.[127–129] Those comparison studies do not identify a clear "winner," i.e., a technique that is significantly better than another; apparently an approach that combines existing algorithms performs better than any individual technique alone.[129] The different definitions given to the similarities of two structures are reviewed below, and two methods for protein structure alignment are described, one based on distance matrices (DALI),[8,130] and one based on dynamic programming and comparison of structures in coordinate space (STRUCTAL).[118,119] Recent progress in developing a closed-form protein structure alignment algorithm is then described.

Scoring Functions for Protein Structure Superposition

Because the concept of "optimal" correspondence is ambiguous, the protein structure superposition problem is not uniquely defined. Instead, finding the best superposition of two proteins corresponds to a family of optimization problems, which are specified by the weight given to the similarity (preferably a small deviation between the two subsets), and the correspondence length (preferably large).

Various measures of similarity between two sets of points exist. In the section on rigid-body transformation, the cRMS value, which measures

the root mean square deviation between the coordinates of the points of the two sets, was described. For a given correspondence length, the cRMS can be minimized with a closed-form algorithm (see above). When both the cRMS and the correspondence length must be optimized, no closed-form solutions are known. Approximate solutions that are usually based on heuristics do not minimize the cRMS directly, as a consequence of being very sensitive to outliers (the definition of cRMS, given in Eq. [2], shows that it is a sum of the Euclidian distances between the points of the two structures; if a few of these points are far apart, they will have a significant impact on the value of cRMS). An example of a solution to this outlier problem was given by Levitt et al.[118,119] who introduced a scoring function with a Lorentzian shape:

$$ST(P,Q) = \max_{R,T} \sum_{i=1}^{p} \frac{20}{1 + \| a_{pi} - Rb_{qi} - T \|^2} - 10G_{P,Q} \qquad [33]$$

The summation extends over the length of the correspondence between *A(P)* and $B(Q)$, $G_{P,Q}$ is the total number of gaps in $A(P)$ and $B(Q)$, and R and T are the optimal rotation and translation that maximize the score (as opposed to reaching a minimum for cRMS, as seen in Eq. [1]).

An alternative measure of protein structure similarity is the distance root mean squared deviation (dRMS) that compares corresponding internal distances in the two sets of points:

$$dRMS = \left[\frac{2}{p(p-1)} \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} M\big(\| a_i - a_j \|, \| b_i - b_j \|\big) \right]^{1/2} \qquad [34]$$

where p is the cardinality of the two sets.

There is no consensus in the scientific community on the definition of the metric M that should be used to compare the two internal distances $\|a_i - a_j\|$ and $\|b_i - b_j\|$. When comparing two pairs of atoms between two structures, Taylor and Orengo[101] defined a distance or similarity score in the form $e/(D+f)$, where D is the difference between the two intramolecular distances and e and f are arbitrarily defined constant values. Holm and Sander[8] defined a similarity score as $(e - [D/\langle D \rangle])\exp(-[\langle D \rangle / f]^2)$, where $\langle D \rangle$ is the average of the two intramolecular distances. Rossmann and Argos[131] and Russell and Barton[132] used a score defined as $\exp(-[D/e]2)\exp(-[S/e]2)$, where S takes into account local neighbors for each pair of atoms. Currently, no clear evidence exists as to which score performs best.

All scoring functions use geometry for the comparison and ignore similarities in the environment of the residues. Suyama et al.[133] proposed another approach in which they ignored the 3-D geometry altogether and compared

structures on the basis of their 3-D profiles[134] alone, using dynamic programming. These profiles include information on solvent accessibility, hydrogen bonds, local secondary structure states, and side-chain packing. Although this method can align two-domain proteins with different relative orientations of the two domains, it often generates inaccurate alignments.[133] Jung and Lee[135] improved on this method by iteratively refining the initial profile alignment with dynamic programming and 3-D superposition. Their method, which they call SHEBA, was found to be fast and as reliable as other alignment techniques (although it was only tested on a small number of protein pairs). Kawabata and Nishikawa[136] derived a novel scoring scheme for generating structural alignments based on the Markov transition model of evolution. The similarity score between two structures i and j is defined as $\log(P(ji)/P(i))$, where $P(ji)$ is the probability that structure j changes to structure i during evolution and $P(i)$ is the probability that structure i appears by chance. The probabilities are estimated with a Markov transition model that is equivalent to the Dayhoff's substitution model for amino acids. Three types of scores were considered: (1) a score based on accessibility to solvent; (2) a residue-residue distance score; and (3) an SSE score.

Superposition Based on Internal Distance Matrices: DALI

Associated with every protein chain A of n atoms is an $n \times n$ real symmetric matrix D, where $D(i, j)$ is the Euclidian distance between atoms i and j of A. This matrix is the "internal distances matrix" of A, also called the distance map of A. The two representations of a protein (by the coordinates of its atoms and by its internal distances matrix) are closely related. Calculating the distance matrix from the coordinates is easy and takes quadratic time in n. Conversely, it is known that the atomic coordinates of a protein can be recovered from the distances matrix, using the method of distance geometry.[137,138] The recovered atomic coordinates are the original ones, modulo a rigid transformation (and possibly a mirror transformation). This equivalence between coordinates and internal distances has led to two different measures of protein similarities, each based on one of the two representations. The use of the internal distances to compare protein structures has a major advantage of by passing the need to find an optimal rigid transformation that superposes the two structures. As a consequence, most algorithms have been developed that compare the internal distances matrix to align protein structures, the most popular being the Distance ALIgnment algorithm (DALI), which is briefly described below.

Holm and Sander[8] developed a two-stage procedure in DALI that uses simulated annealing to build an alignment of similar hexapeptide backbone fragments between two proteins.

In the first stage, the two protein structures to be compared are divided into overlapping hexapeptides. A contact map, which contains all internal distances, is generated for each hexapeptide. Although residues in the proteins

belong to several overlapping hexapeptides, they are assigned to the hexapeptide with the closest contacts to other fragments. Contact maps of the two proteins are matched by comparing their internal distances, using an "elastic" score of the form $(e - [D/\langle D\rangle])\exp(-[\langle D\rangle/f]2)$, where D is the difference between the two distances to be compared, $\langle D\rangle$ is the average distance, and e and f are parameters. This score is less sensitive to distortion for long-range distances. For the sake of efficiency, only hexapeptide pairs having a similar backbone conformation are compared. Hexapeptides whose contact maps match above a given threshold are stored in lists of fragment equivalences.

In the second stage, an optimization protocol, based on simulated annealing, explores different concatenations of the equivalent hexapeptide pairs. Similarity is assessed by comparing all distances between the aligned substructures. Each step of this second stage consists of addition, replacement, or deletion of residue equivalences (in units of hexapeptides). Because hexapeptides can overlap, each step results in the addition of one to six residues. Once all candidate hexapeptide pairs have been tested, the alignment is processed to remove fragments with a negative contribution to the overall similarity score.

This two-stage method, implemented in DALI,[130] has been used to compare representatives from all nonhomologous (in sequence) families in the protein data bank.[3,4] More details are given in the DALI Domain Classification below.

Superposition Based on cRMS: STRUCTAL

The internal distances matrix is invariant under rigid and mirror transformations of the protein. Although this lead to a simplification of the protein structure superposition problem because algorithms that compare proteins based on internal distances do not need to find the optimal rigid transformation, mirror transformation may introduce errors because mirror images (such as a right-handed helix versus a left-handed helix) will not be detected as being different. Consequently, in addition to the approaches based on the internal distances matrix, methods have been concurrently developed to solve the protein structure alignment problem using coordinates to measure the similarity between proteins. These methods are based on heuristic algorithms that optimize simultaneously the correspondence between the two proteins and the rigid transformation. An example is the algorithm developed by Subbiah et al. and implemented in the program STRUCTAL.[118]

STRUCTAL starts with an arbitrary equivalence of atoms between the two proteins A and B. This equivalence defines a list of corresponding residues (represented by their C_α atoms) that are superimposed with the optimal rigid-body transformation. Once the two proteins are superimposed, the program computes a structure alignment matrix SA. $SA(i,j)$ measures the similarity between residue i of protein A and residue j of protein B, based on a function

of the distance d between $C\alpha_i$ and $C\alpha_j$, after optimal superposition. This function is defined such that

$$SA(i,j) = \frac{20}{1 + d(C\alpha_i, C\alpha_j)^2/5} \quad [35]$$

It is simple to compute, and it has the important properties of being positive and of decreasing monotonically with increasing distances. A new alignment is then determined by searching the distance matrix for the alignment with the best score. Dynamic programming rapidly [$O(n^2)$ operations] finds the optimum for the given structure alignment matrix and for a given gap penalty. In STRUCTAL, the gap penalty is set constant, equal to 10. The new alignment leads, in turn, to a new set of equivalencies between the proteins; this set is then used to re-superimpose the two proteins in three dimensions giving rise to a new structure alignment matrix. The procedure is iterated until the alignment matrix does not change with additional iterations.

Because this structural alignment procedure is based on dynamic programming and is iterative, the results depend on the initial equivalence assigned. To account for this potential problem, STRUCTAL starts with five different equivalences. The first three equivalences are simple, corresponding to aligning the chain beginnings, the chain ends, and the chain midpoints of the two structures, respectively, without allowing any gaps. The fourth choice maximizes the sequence identity of pairs of residues considered equivalent, whereas the fifth choice is based on the similarity of the C_α torsion angles between the two chains. After repeating the iterative scheme to find the optimal equivalence and superposition for each of the five initial sets of equivalences, the optimal alignment is chosen as the one with the highest score. Extensive studies have shown that not one of the five initial sets works better than another in converging to the highest score.[119]

An Approximate Polynomial Time Algorithm

A prevailing sentiment among scientists developing algorithms for protein structure alignment is that structure comparison requires exponential computer resources, and so development should focus on heuristic approaches. Consequently, there are no guarantees of finding an optimal alignment with respect to any scoring function with any of the existing methods. In addition, if one of these methods fails to find a satisfactory alignment, it cannot be ruled out that a good alignment exists. There is one interesting, albeit theoretical, exception to this. Kolodny and Linial[139] have developed a polynomial-time algorithm that optimizes simultaneously the correspondence and the rigid transformation leading to a structural alignment. The computation cost of their algorithm is of the order $O(n^{10})$, and as such, it is not practical to implement. This algorithm, however, is not heuristic: it guarantees finding ε-approximations to all solutions of the protein superposition problem, where

these solutions correspond to maxima of the STRUCTAL score *ST* defined in Eq. [35].

For an alignment algorithm to be polynomial, the following conditions must hold:

1. Given a rigid transformation, it should be possible to find an optimal correspondence in polynomial time.
2. The number of rigid transformations under consideration must be bounded by a polynomial.

The STRUCTAL score *ST* is amenable to dynamic programming and therefore can be used to find an optimal correspondence in time and space requirements of order $O(n^2)$, for any given rigid transformation r. The score of this optimal correspondence is denoted as $ST_{opt}(r)$. It validates condition 1. The validity of condition 2 is derived from a lemma given by Kolodny and Linial, which states that for all ε, a finite set $G = G(\varepsilon)$ of rigid transformation exists, such that for every choice of a rigid transformation r, a transformation r_G in $G(\varepsilon)$ exists such that $\| ST_{opt}(r) - ST_{opt}(rG) \| < \varepsilon$, and cardinal$(G) = |G|$ is polynomial in n.

This lemma suggests the following algorithm for the structural alignment problem. For a given value of ε, build $G(\varepsilon)$, the discrete sampling of the space of rigid transformation, and evaluate ST_{opt} over all rigid transformations in $G(\varepsilon)$. The ε-optimal structure alignments of the two proteins are guaranteed to be within ε of the maxima found in the exhaustive search over $G(\varepsilon)$. A major advantage of this exhaustive algorithm is that if it fails to find a good alignment, no good alignment exists. Because the size of $G(\varepsilon)$ is of the order $O(n^{10}/\varepsilon^6)$, the computing time required by this algorithm is still prohibitive. As such, the contribution of Kolodny and Linial should be viewed as mostly theoretical, rather than as practical, but it does provide insights about the complexity of protein structure alignments.

cRMS: An Ambiguous Measure of Similarity

The root mean square deviations (cRMS or dRMS) remain the measures of choice by structural biologists to describe the similarity between two proteins, even though most algorithms for protein structure alignments use scoring schemes that differ significantly from simply taking into account interatomic distances (see above). Both cRMS and dRMS are based on the L_2-norm (i.e., the Euclidian norm), and as such, they suffer from the same drawback as the residual χ^2 in least-squares minimization in which the presence of outliers introduces a bias in the search for an optimal fit and the final measure of the quality of the fit may be artificially poor because of the sole presence of those outliers. Another problem of relying on RMS deviations is that it does not always satisfy the triangular inequality. More precisely, the triangular inequality is satisfied when the correspondences between the

proteins always involve the same points.[140] Generally by varying correspondences, it is easy to find a situation in which the triangular inequality is not satisfied. Consider, for example, two proteins A and B that are dissimilar, and the two-domain protein C, whose subdomains C1 and C2 are very similar to A and B, respectively. In this example, the RMS deviations between A and C and between B and C are low, but the RMS deviation between A and B is large, violating the triangular inequality that states that RMS(A, B) $\leq$ RMS(A, C) + RMS(C, B). As a consequence of these limitations, RMS is a useful measure of structural similarity for only closely related proteins.[141] Several other measures have therefore been proposed to circumvent these problems. The STRUCTAL score S2 (Eq. [34]) was shown to be a more reliable indicator of structure similarity than RMS because it depends on the best-fitting pairs of atoms (thereby removing the weights of outliers). In contrast, RMS gives equal weight to all pairs of atoms. Lesk[142] recently proposed replacing the L_2-norm in the RMS definition by the L norm, also called the Chebyshev norm, to yield the new score:

$$S = \max_{i \in [1,N]} \{ \| x(i) - y(i) \| \} \qquad [36]$$

S reports the worst-fitting pair of atoms (after optimal superposition of the two structures) and, as such, is even more sensitive to outliers than is the RMS. Yang and Honig[120] defined a new protein structure similarity measure, the protein structural distance (PSD). PSD combines a secondary structural alignment score and the RMS deviation of topologically equivalent residue pairs. It thus incorporates the resolution power of both RMS for closely related structures and the secondary structure score for proteins that can be very different. By analyzing the PSD scores obtained from more than one and a half million pairs of proteins, Yang and Honig[120] proposed that a continuum of protein conformation space exists, conflicting with existing ideology in structural classification databases such as SCOP (Structural Classification Of Proteins[6]) and CATH (Class, Architecture, Topology and Homologous Superfamilies[7]). May[143] assessed 37 different protein structure similarity measures for their ability to generate accurate clusters in a hierarchical classification of 24 protein families. It was found that the sum of ranks of distances at aligned positions was a better measure of similarity than the direct sum of distances and that RMS computed over the subset of core-aligned positions performs better than normal RMS computed over all-aligned positions. Variations in the hierarchical classification of protein structures brings into question the validity not only of the measure used for the clustering, but also of the hierarchical clustering. The difficulty associated with defining a similarity score for protein structures reflects the fact that most questions related to structure comparison do not have a unique answer[144–146] and brings to the fore that the problem is ill posed, requiring additional information to provide a well-defined solution. As

an example, consider fold recognition applications, where predictors focus on the well-conserved core region of the protein and pay less attention to the loop geometry. In such cases, it makes sense to define a similarity score that includes only atoms in the core.

Having a quantitative measure of protein structure similarities is essential when assessing the quality of protein structure predictions, such as those generated for the Critical Assessment of techniques for protein Structure Prediction (CASP) project, organized in the form of a meeting held in alternating years at Asilomar, California. For the special case of comparing a predicted structure with its experimental counterpart, the equivalence list is known because the two sequences are identical, which thus reduces the complexity of the problem. On the other hand, each structure prediction may omit different residues depending on how the prediction algorithm works and different parts of the predicted structure may omit geometries of variable quality. Hubbard[147] avoided the problem by generating many superpositions and by calculating the best RMS for each set of equivalent residues (not necessarily contiguous), to provide an RMS/coverage graph, which evaluated predictions at CASP3. The RMS/coverage plot can also be interpreted as defining the number of equivalent residues for a given RMS value.

Differential Geometry and Protein Structure Comparison

The inherent problems of RMS as a measure of protein structure similarities, and the difficulties encountered by the existing heuristic algorithms whose aim is to solve the protein structure superposition problem, have led to the development of a new approach for comparing protein structure, based on differential geometry and the concept of protein shape descriptors. A tutorial on molecular shape descriptors can be found in a previous volume of this series.[148] The idea behind this approach is simple: Represent the protein structure with a vector of geometric properties (GPs) such that the comparison of two protein structures is performed by comparing their GP vectors, usually with a Euclidian metric. Once the GP vectors have been computed, structure comparison with this scheme becomes instantaneous, and it can then be performed over entire databases. The success of this approach depends on the quality of the geometric properties included in GP and on their ability to uniquely capture the geometric properties of the protein. Because there is a growing interest to define such protein shape descriptors, two descriptors derived from knot theory, the writhe and the radius of curvature of a polygonal curve, are reviewed here.

The Writhe of a Protein Chain

The writhe of a polygonal curve is the signed average crossing number of the curve, where the average is taken over the observer's positions, located in all space directions.

Consider a polygonal curve A defined by N line segments i. The writhe of A is computed according to

$$Wr(A) = I_{1,2}(A) = \sum_{0<i_1<i_2<N} W(i_1, i_2) \quad [37]$$

with

$$W(i_1, i_2) = \frac{1}{2\pi} \int_{t_1=i_1}^{i_1+1} \int_{t_2=i_2}^{i_2+1} w(t_1, t_2) dt_1 \; dt_2 \quad [38]$$

where $W(i_1, i_2)$ is the contribution to the writhe of line segments i_1 and i_2. $W(i_1, i_2)$ is the probability of seeing the line segments cross when viewed from an arbitrary direction, multiplied by the sign of the crossing. Computation of $W(i_1,i_2)$ is described in Figure 8. Similarly, the unsigned average number of crossing, usually referred to as the average crossing number, is given by

$$I_{|1,2|}(A) = \sum_{0<i_1<i_2<N} |W(i_1, i_2)| \quad [39]$$

A whole family of structural measures can be envisaged with $W(i_1, i_2)$ and $|W(i_1, i_2)|$ as building blocks,[149] such as

$$I_{(1,3)|2,4|} = \sum_{0<i_1<i_2<i_3<i_4<N} W(i_1, i_3) |W(i_2, i_4)| \quad [40]$$

and

$$I_{(1,4),(2,6),(3,5)} = \sum_{0<i_1<i_2<i_3<i_4<i_5<i_6<N} W(i_1, i_4) W(i_2, i_6) W(i_3, i_5) \quad [41]$$

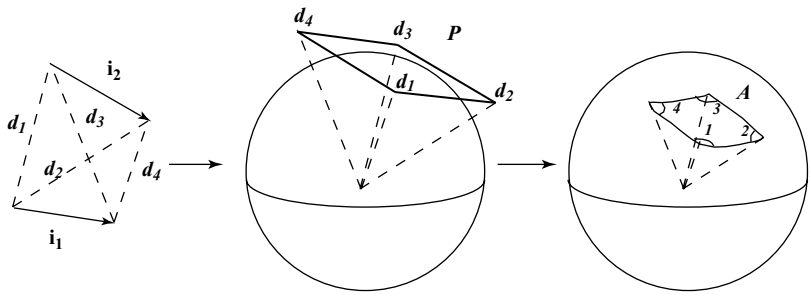


Figure 8 Computing $W(i_1, i_2)$, the writhe of two segments i_1 and i_2. (Left panel) The endpoints of the segments i_1 and i_2 are connected by 4 vectors d_1, d_2, d_3, and d_4 that generate a parallelogram P of directions (center panel). The area A of the projection of P on the surface of the unit sphere is the segment-segment writhe $W(i_1, i_2)$. A is computed as: $A = (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - 2\pi$ (right panel).

These measures are inspired by Vassiliev knot invariants.[150] They form a natural progression of curve descriptors, much as moments of inertia and their correlations define solids.

The writhe and the average crossing number have been used extensively to characterize DNA molecules and more specifically supercoiled DNAs.[151,152] They have also been used to describe proteins. Levitt[153] has used writhe to distinguish different chain threading. Arteca et al. used the writhe as a protein shape descriptor.[154–158] Rogen and Bohr[149] have used the writhe, the average crossing number, and their higher order correlations to define a feature vector that characterizes protein structures. More recently, Rogen and Fain[159] have compared protein structures with feature vectors in $\Re^{30}$ similar to those defined by Rogen and Bohr, using a pseudo-metric, which is the Euclidian distance between the feature vectors. That pseudo-metric is the scaled Gauss metric (SGM) (this name was chosen as the writhe of a continuous curve and is usually computed with a Gauss integral[160]). Rogen and Fain[159] show that SGM performs extremely well as a protein structure classifier, using both CATH and SCOP as test sets. Because both CATH and SCOP include all protein chains in the PDB, they are highly redundant and cannot be considered as discriminative benchmarks. Despite this reservation, the results of Rogen and Fain provide a new way to compare and classify protein structures, using geometric protein shape descriptors.

Protein Chain Thickness and Generalized Radius of Curvature

Any smooth, nonintersecting curve can be thickened to a smooth, nonintersecting tube of constant radius centered on the curve. If the curve is a straight line, there is no upper bound for the radius, but for any other curve, there is a critical radius above which the tube ceases to be smooth or shows self-contact. This critical radius is referred to as the thickness of the curve, and it is used as a shape descriptor in knot theory. Because the geometry of DNA and protein molecules is characterized by the geometry of their backbone chain, it is natural to check whether thickness could be used as a shape descriptor for these molecules. Gonzalez and Maddocks[161] introduced the concept of generalized radius of curvature and used it to characterize the geometry of DNA molecules. Their definition of generalized radius of curvature is based on the fact that any three noncollinear points x, y, z in 3-D space define a unique circle whose radius is given by

$$r(x,y,z) = \frac{|x-y||x-z||y-z|}{4A(x,y,z)} \qquad [42]$$

where $A(x,y,z)$ is the area of the triangle with vertices at x, y and z and $|x-y|$ is the Euclidian distance between x and y. Let us consider a discrete curve C,

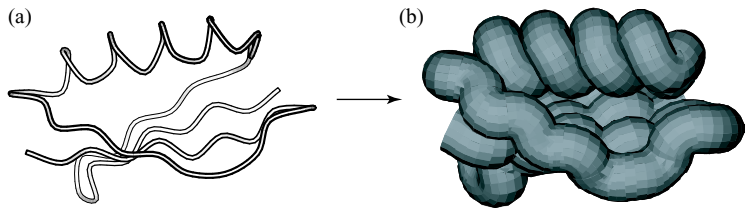


Figure 9 Thickness of a protein. (a) The structure of the B1 immunoglobulin-binding domain of streptococcal protein G is visualized as a thin tube. (b) View of the same tube inflated to its "thickness" (i.e., to a radius above which the tube ceases to be smooth, or shows self contact). Note that no free space exists between consecutive turns of the helices. Figure 9a drawn with MOLSCRIPT[17] and 9b with VMD.[18]

defined by n nodes (c_1, c_2, ... c_n). Gonzalez and Maddocks define the generalized radius of curvature of C at c_i by

$$\rho_C(c_i) = \min_{\substack{1\le j,k\le n \\ i\neq j\neq k\neq i}} r(c_i, c_j, c_k) \qquad [43]$$

$\rho_C(c_i)$ is the radius of the smallest circle passing by c_i and two other distinct nodes of C. $\rho_C(c_i)$ should be distinguished from the local radius of curvature ρ defined at c_i by $\rho(c_i) = \rho(c_{i-1},c_i,c_{i+1})$. The thickness $\Delta(C)$ of the discrete curve C is related to the generalized radius of curvature by

$$\Delta(C) = \min_{1\le i\le n} \rho_C(c_i) \qquad [44]$$

In other words, $\Delta(C)$ is the radius of the smallest circle passing by three points of C.

Figure 9 illustrates the "thickness" of a small globular protein. The concepts of thickness and generalized radius of curvature have been used to characterize the geometry of DNA.[151,152] They have also been used as a "potential" that captures the geometry of a protein,[162–166] which was used, for example, in protein structure prediction computer experiments.[167] Thickness and generalized radius of curvature have not yet been used for protein structure comparison, but it is expected that they would prove useful for detecting protein structure similarities, especially when combined with other features such as writhe.

Upcoming Challenges for Protein Structure Comparison

The most difficult application of protein structure comparison comes in classifying known protein structures into different clusters corresponding to fold families. The role of such classifications is to organize structure databases such as the PDB, in hopes of detecting similarities at the structure level that

cannot be detected at the sequence level, and more generally, to detect evolutionary relationships between proteins. The existing protein structure classification schemes are reviewed in the next section. Multiple challenges must be overcome by a protein structure comparison program in this application. First, it must deal with different levels of structural similarities, it must identify similarities even when those similarities form a small proportion of the proteins being compared, and it must handle insertions of arbitrary size as well as permutations of substructures. Second, it must deal with the fact there may be more than one acceptable solution for the structural alignment of two proteins. These multiple, equivalent solutions (in terms of cRMS and length of the equivalence) may all be viable from a biological perspective,[145] and therefore cannot be ignored. Third, the size of most protein structure databases has grown exponentially in the recent years, and the growth rate is expected to continue as the structural genomics projects enter their productive phases. A need exists for fast techniques to compare and classify these structures, faster than the existing techniques that are too time consuming to be of use.

None of the existing methods, including those described in length above, propose solutions to meet these challenges. Heuristic methods were developed for the sake of efficiency; yet no guarantee exists that they can find the optimal superposition. Also, some of these heuristic methods cannot detect alternative, equally acceptable solutions. The approximate solution developed by Kolodny and Linial[139] resolves some of these issues in the sense that it can detect all maximal solutions with an ε of the optimal solutions, but its computing cost (of the order of $O(n^{10}/\varepsilon^6)$ where n is the size of the proteins considered) makes it unsuitable for large-scale comparisons. There is a need to develop faster, more robust, and exhaustive approaches to solving the myriad of problems associated with protein structure comparison. This field in fact remains an active area of development in structural biology, but solutions may come from interdisciplinary research groups. The problem of comparing two protein structures can be reformalized as the problem of comparing two sets of points in 3-D space. As such, it can be seen as a problem of computational geometry, and it is expected that collaboration between structural biologists well versed in deciphering protein structures and computer scientists who focus on geometric problems could provide the synergy required for significant progress. The recent advances in the application of differential geometry to protein structure (see the sections on writhe and curve thickness above) are signs that these collaborative efforts are working.

PROTEIN STRUCTURE CLASSIFICATION

Perutz et al.[30] showed in 1960 that myoglobin and hemoglobin, the first two protein structures to be solved at atomic resolution using X-ray crystallography, have similar structures even though their sequences differ. These two

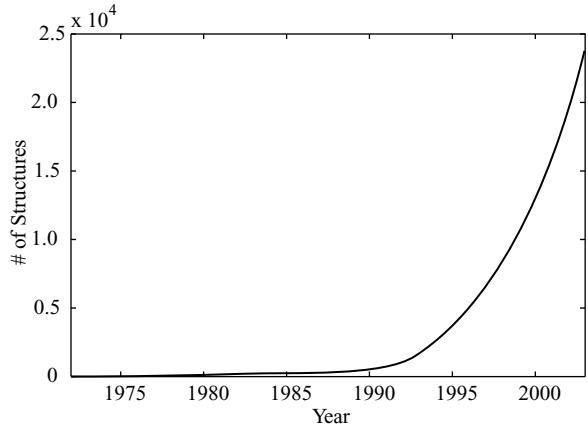


Figure 10 Statistics on the PDB. The number of structures (proteins and nucleic acids) available in the Protein Data Bank (PDB)[3,4] is plotted against time, starting from 1973 when the PDB was created.

proteins are functionally similar, as they are involved with the storage and the transport of oxygen, respectively. Since then, there has been a continued interest in finding structural similarities between proteins, in the hope of revealing shared functionality that could not be detected by sequence information alone. A logical consequence of this interest is the development of systems for classification of protein structures that identify and group proteins sharing the same structure so as to reveal evolutionary relationships. Classifying protein structures has now become essential because of the volume of structural data available (see Figure 10). In parallel with the development of protein structure classification methods are the developments for many classifications of protein sequences described in length in Ref. 168. Table 6 lists some resources available for sequence classification.

All current structural classification methods are based on the same scheme: Protein structures are first divided into discrete, globular domains, which are then classified at the levels of (1) "class," (2) "folds," (3) "superfamilies," and (4) "families." The differences among existing schemes come from the methods that define the domains and the procedures that classify. After reviewing the terms that define a classification, the three main protein structure classifications available, SCOP, CATH, and the DALI Domain Dictionary (DDD), will be described. Links to these databases and related services are listed in Table 7.

The first complication associated with structure classification involves the fact that protein structures are often composed of distinct globular domains. Because these domains can function individually, with distinct functional roles, proteins are usually separated into domains before classification.

Table 6 Resources for Classification of Protein Sequences

Scheme	Description	Web Access
PFam	Domain-level classification of protein sequences	http://www.sanger.ac.uk/Software/Pfam/
PRINTS	Fingerprints information on protein sequences	http://www.bioinf.man.ac.uk/dbbrowser/PRINTS/
PROSITE	Sequence motif definition	http://www.expasy.org/prosite/
TIGRFAMS	Protein family database	http://www.tigr.org/TIGRFAMs/
PRODOM	Protein domain database	http://protein.toulouse.inra.fr/prodom.html
BLOCKS	Multiple-alignment blocks	http://blocks.fhcrc.org/
eMOTIF	Protein motif database, derived from PRINT and BLOCKS	http://motif.stanford.edu/emotif/
CluSTr	Clusters of related proteins	http://www.ebi.ac.uk/clustr/
COGS	Clusters of orthologous groups	http://www.ncbi.nlm.nih.gov/COG/
ProtoMap	Hierarchical classification of protein sequences	http://protomap.cornell.edu
TRIBES	Protein family databases	http://maine.ebi.ac.uk:8000/services/tribes/
PIR international	Protein sequence databases	http://pir.georgetown.edu/
SYSTERS	Protein family database	http://systers.molgen.mpg.de/
SMART	Small motif database	http://smart.embl-heidelberg.de/
UniProt	Catalog of information on proteins	http://www.expasy.uniprot.org/
InterPro	Databases of protein families and domains	http://www.ebi.ac.uk/interpro/

How to identify and delineate these domains is still an open problem as discussed above. It is important to realize that the existing algorithms for domain identification do not always agree; the corresponding discrepancies in domain definition translate into differences between structural classifications that do not share the same definition.

Table 7 Resources for Protein Structure Classifications

Scheme	Description	Web Access
SCOP	Structural Classification of Protein: manual	http://scop.mrc-lmb.cam.ac.uk/scop/index.html
CATH	Class, Architecture, Topology, Homology: semiautomatic classification of proteins	http://www.biochem.ucl.ac.uk/bsm/cath
DALI Fold Classification	Automatic classification of DALI domain using Dali. Supersedes FSSP	http://www.bioinfo.biocenter.helsinki.fi:8080/dali/index.html
ASTRAL	Databases and tools for analyzing protein structure; derived from SCOP	http://astral.berkeley.edu/
HOMSTRAD	Aligned 3-D structures of homologous proteins	http://www-cryst.bioc.cam.ac.uk/data/align

Once proteins are divided into domains; the domains are then classified hierarchically. At the top of the classification we usually find the "class" of a protein domain, which is generally determined from its overall composition in secondary structure elements. Three main classes of protein domains exist: mainly α domains, mainly β domains, and mixed $\alpha - \beta$ domains (the domains in the $\alpha - \beta$ class are sometimes subdivided into domains with alternating α/β secondary structures and domains with mixed $\alpha + \beta$ secondary structures). In each class, domains are clustered into "folds" according to their topology. A fold is determined from the number, arrangement, and connectivity of the domain's secondary structure elements. The folds are subdivided into "superfamilies." A superfamily contains protein domains with similar functions, which suggests a common ancestry, often in the absence of detectable sequence similarity. Sequence information defines "families," i.e., subclasses of superfamilies that regroup domains whose sequences are similar.

Classification schemes are designated as being curated and automatic. A curated classification is one that is based mainly on human expertise, sometimes guided by computer analyses, to identify similarities between protein structures for organization into groups. An automated classification relies exclusively on the results of a computer procedure to identify the similarities, which are subsequently processed automatically to generate the groups. One advantage of curation is the typically high quality of the clustering results; the disadvantage is that curation is difficult to scale to high volumes of data. Conversely, automatic procedures are fully reproducible and scalable, but they may inaccurately assign similarity. The three most common protein structure classifications illustrate these differences: SCOP is almost completely manually derived, the DALI domain dictionary is based on a fully automated procedure, and CATH is intermediate between these two classification, using automated procedures complemented with human interventions.

The Structure Classification of Proteins (SCOP)

SCOP[6] is a repository that organizes protein structures hierarchically to reflect both structural and evolutionary relatedness. SCOP has been constructed manually, from the delineation of the domains in multidomain proteins to the organization of the levels of the hierarchy. It relies on visual inspection and comparison of protein structures, with the assistance of some automatic computer tools to make the task manageable and to help provide consistency and generality. Since its creation in 1994, SCOP has been updated regularly, with an average frequency of two releases a year. The latest update of SCOP, 1.65, was built from the 20,619 PDB entries (54,745 domains) available on August 1, 2003 and was released in December 2003. Statistics on the growth of SCOP are given in Figure 11.

SCOP is a hierarchic classification with four major levels: classes, folds, superfamilies, and families. As recognized by the authors of SCOP, the exact

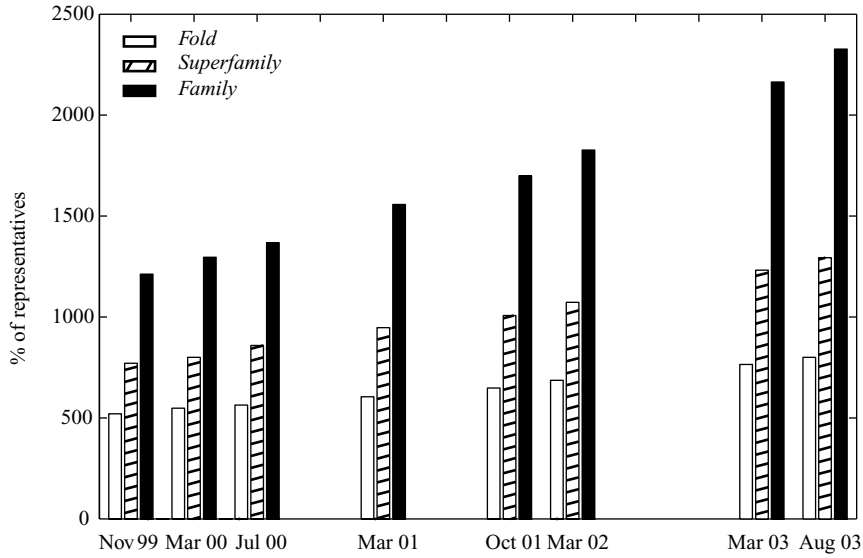


Figure 11 Statistics of the SCOP classification of proteins. The numbers of folds, superfamilies, and families in SCOP are plotted against "time", where time is the timestamp of the PDB used to generate the update of SCOP.

positions of boundaries separating these levels are subjective; where any doubts of similarity existed, they have chosen to create new divisions at the family and superfamily levels.

At the top of the SCOP hierarchy are 11 different classes: alpha, beta, alpha and beta (α/β), alpha plus beta ($\alpha + \beta$), multidomain proteins, membrane and cell-surface proteins, small proteins, coiled coil proteins, low-resolution protein structures, peptides, and designed proteins. Note that only the first seven classes are true classes. The remaining ones serve as place holders for protein domains that cannot (yet) be classified among the major classes and are maintained in SCOP for the sake of completeness and compatibility with the PDB.

In each SCOP class, proteins are clustered into groups based on their structure similarity. Each cluster is referred to by SCOP as a fold. Proteins share a common fold if they have the same major secondary structures in the same arrangement and with the same topological connections. Proteins with the same fold may differ at the level of their peripheral elements, which can include secondary structures and turn regions. Note that these peripheral elements can represent up to 50% of the structure. Proteins catalogued together in the same fold may have no common evolutionary origin.

SCOP superfamilies identify probable common evolutionary origin. Proteins whose sequences have low similarities, but that share the same fold and have similar functions, suggest that a common evolutionary origin is probable.

Proteins clustered together into families are clearly evolutionary related. The sequences of two proteins of the same family often have a residue identity greater than 30%. In some cases, a high sequence identity is not needed to affirm common origin; many globins, for example, form a family, even though some members of that family have a sequence identity of only 15%.

The CATH Classification

CATH[7] is a protein structure classification in which protein domains are clustered at four major levels: class (C), architecture (A), topology (T), and homologous superfamily (H), each of which are described below. CATH uses a semiautomatic classification procedure that filters out nonprotein, models, and "C_α-only" structures from the PDB. Only crystal structures solved to resolution better than 3.0 Å are considered, together with all NMR structures. The latest update of CATH, v2.5.1, was released January 28, 2004 and includes 48,391 domains.

Multidomain proteins are subdivided into individual domains with a consensus procedure based on three algorithms for domain recognition: DETECTIVE,[82] PUU,[77] and DOMAK.[77] When all three algorithms generate the same result on a multidomain protein, the common solution delineates the domains of that protein. This consensus procedure resulted in 53% of the proteins included in the CATH release 2.5.1 to be subdivided into domains automatically. The remaining structures were assigned domains manually, using one of the assignments made by the automatic procedure, an assignment obtained from the literature, or based on a new assignment defined by visual inspection.

CATH includes four classes (C): alpha, beta, alpha and beta, and few secondary structure (FSS). The alpha–beta class includes both alternating alpha/beta structures and alpha + beta structures, originally defined by Levitt and Chothia.[31] The class of a protein domain is determined according to its secondary structure composition and packing. Ninety percent of the protein domains were automatically assigned to their class in CATH 2.5.1, using the method developed by Michie et al.[169,170] The remaining 10% of domains were assigned to a class by visual inspection.

The architecture (A) level included in CATH describes the overall shape of the domain structures, as determined by the orientation of their secondary structures, ignoring their connectivity. It is assigned manually. This level has no equivalent in SCOP.

Domains are grouped into topologies (T), or fold families, according to their overall shape and the connectivity of their secondary structures. This is done with the structural alignment program SSAP.[101] Proteins belonging to the same class are compared systematically with SSAP, and the corresponding scores are stored in a two-dimensional matrix. Structure pairs that have a sufficiently high SSAP score (>70) are merged into fold families, using single linkage clustering (for a brief description of this clustering technique, see the Appendix).

The Homologous Superfamily level, or H level, groups together protein domains thought to share a common ancestor. This level is equivalent to the superfamily level defined in SCOP. CATH also includes a Sequence Families level (S-level) that is equivalent to the family level of SCOP.

The DALI Domain Dictionary (DDD)

The DDD, also called the DALI domain classification, is derived from a fully automated method of defining and classifying domains.[171,172] DALI domains are defined by a version of the PUU algorithm[77] that has been updated to consider the recurrence of putative domains.[173] When comparing two protein structures, DALI computes a similarity measure, or S score. The mean and standard deviations of the S scores obtained over all pairs of proteins are evaluated. Shifting the S scores by their mean and rescaling by the standard deviation yield the statistically meaningful Z-scores.

The program DALI was used initially to create the Families of Structurally Similar Proteins (FSSP) database.[174] In FSSP, pair-wise structural comparisons are made between proteins of a representative set, in which no two proteins have greater than 25% sequence identity. For each member of the representative set, a file is created that contains all pair-wise structural matches with a Z-score greater than 2.0. The same procedure generates a complete classification of all protein domains in the PDB90 database, the DDD.[172] PDB90 is a representative subset of the PDB, where no two chains share more than 90% sequence identity. An average linkage hierarchical clustering technique (see the Appendix) generates a fold tree covering the PDB90 database. The pair-wise structural alignments are divided with Z-score cutoffs of 2, 4, 8, 16, 32, and 64, creating a six-character index for each domain. The first level ($Z > 2$) is used as an operational definition of folds. Lower levels should not be confused with the superfamily and family levels of CATH and SCOP, as they are not based on direct functional or evolutionary relationships. Both FSSP and the DDD are continuously updated; this is possible as they are both derived from a fully automated procedure.

Comparing SCOP, CATH, and DDD

SCOP, CATH, and DDD agree on most of their classifications, despite differences in the classification methods they have implemented, and in the rules of protein structure and taxonomy they are based on. Hadley and Jones[175] were the first to publish a detailed comparison of the fold classifications produced by SCOP, CATH, and FSSP. They showed that the three classification systems tend to agree in most cases, and that the discrepancies and inconsistencies are accounted for by a small number of problems. Among these, the domain assignment plays a crucial role. As mentioned, the separation of proteins into domains is a difficult and often subjective process. Many

protein structures are assigned different numbers of chains in SCOP, CATH, and FSSP. An obvious domain problem that results is the exclusion of one part of a protein. Hadley and Jones[175] reported the case of papain, a cysteine proteinase from papaya, which was treated as a single domain by SCOP, leaving the catalytic cysteine, histidine, and asparagines together to form the active site, while CATH split the protein into two domains, separating the cysteine from the asparagine and histidine, and rendering each domain effectively functionless. Note that this difference between SCOP and CATH has been corrected since Hadley and Jones published their study, and papain is now treated as a single domain in CATH. Another discrepancy between the structural classifications originates from the "fold overlap" problem, where a fold within one classification encompasses more than one fold within another classification. When a domain is classified in CATH as a three-layer ($\alpha\beta\alpha$) sandwich Rossmann fold, there are several SCOP folds to which it could belong. Although the structures are geometrically similar, SCOP can separate them to reflect an evolutionary distinction. This "fold overlap" problem is observed, for example, for the protein 1phr, and the chain A of the proteins 1gar and 1lfa, corresponding to a phosphotyrosine protein phosphatase, a formyltransferase, and an integrin, respectively. All three structures contain a three-layer sandwich Rossmann fold and are consequently regrouped in the same Topology in CATH (topology index 3.40.50), while they are representatives of their fold class in SCOP (classes c.44, c.65, and c.62, respectively).

Despite these discrepancies, Hadley and Jones[175] recognized the merits of all three classifications and concluded that no one method is distinctly superior to another. They characterize SCOP as a valuable resource for detailed evolutionary information, CATH as a source of geometric information, and FSSP as a raw source of information, which is continually updated.

Divergences in protein structure classifications have triggered the search for a consensus description of the protein structure space. Day et al.[176] recently repeated the comparative study of SCOP, CATH, and DDD previously done by Hadley and Jones, using updated versions of the classifications. Although Day et al. find significant levels of agreement between the three classifications, they highlight disparities whose origins are similar to those found earlier by Hadley and Jones. To remove these disparities, they introduced the concept of consensus folds. Day et al. start from a nonredundant subset of protein domains. To be considered in the analysis, the authors insisted that 80% of the sequence of a domain in SCOP must be present in a DALI domain definition, 80% of the DALI domain must be present in the SCOP definition, and so on for the other pair-wise combinations of the classification systems. Redundant domains were considered as those having >95% sequence identity to a previously counted domain. Each domain in the nonredundant subset was assigned a fold identifier, which corresponds to its classification in SCOP, CATH, and DDD. Domains were then clustered on the basis of their fold identifiers, and the corresponding clusters were referred

to as metafolds. The nonredundant set contained 5720 domains, clustered into 1130 metafolds. About half of these domains are described by one of the top 30 metafolds. These metafolds represent the consensus information contained in SCOP, CATH, and DDD, and as such define a consensus view of the protein fold space.

CONCLUSIONS

Proteins are key molecules in all cellular functions. Nature has extensively explored their sequences and structures to build the library of functions needed for the diversity of life, taking into account all external constraints and the corresponding adaptation. The wealth of information encoded in the protein sequences and structures therefore provides the clues needed to unravel the mysteries of life and its evolution and adaptation over time. Understanding this diversity has become a key topic in recent genetics and molecular biology studies, catalyzed by the development of numerous genomics and structural genomics projects. More than 220 whole genomes have been sequenced and published on the World Wide Web, and more than 1200 are currently under study, which corresponds to databases in excess of one million nonredundant protein sequences. In parallel, the Protein Data Bank contains structural data on more than 27,000 proteins. The challenge now is to organize these data in a way that evolutionary relationships between proteins can be uncovered and used to understand better protein function. The past few years have seen an explosion of techniques in ''bio-informatics'' for organizing and analyzing protein sequence families. Although such approaches can detect homologous proteins, they usually fail to detect remote homologues, i.e., pairs of proteins that have similar structure and function, but that lack easily detectable sequence similarity. Because protein structures are more highly conserved than are protein sequences, there is a growing interest in studying evolution based on an understanding of the protein structure space. The first steps common to the analysis of any large set of data are to group together data points that are similar, and then to identify connections between those elementary groups. These steps are usually performed with classification techniques. In the case of protein structures, this has led to the construction and maintenance of protein structure classifications, which have been reviewed in this tutorial.

Reliable protein structure superposition remains a bottleneck when carrying out a protein structure classification. Comparing and grouping proteins require a definition of the similarity of two structures. Similarity in structural alignment is geometric and captured by the cRMS deviation of the aligned atoms. Other properties of structural alignments that are likely to be significant are the number of positions matched and the number and length of gaps. Good alignments match more positions, have fewer gaps, and are more similar than do poor alignments. Because these properties of alignments

are not independent (shortening the alignment or introducing many gaps can lower the cRMS), researchers have devised alignment scores that attempt to balance their influences. Several measures of similarity have consequently been developed.[143] Perhaps the most significant recent improvements in this area have been in the protocols for assessing the statistical significance of these measures.[119,177] These statistical measures of similarity are now being used in structure comparison algorithms. Ideally, they should detect reliably distant relatives and be fast enough to scan large databases of representative protein structures. Existing methods have been designed to satisfy one or the other, but not both of these two criteria simultaneously (see the section on protein structure superposition). A need exists for a fast, reliable protein structure superposition program.

Critical to the classification of proteins is the definition of domains. It has long been hypothesized that domains are the important evolutionary units. It is supported by recent analyses of the available genome data, which suggest that at least 60% of the genes are multidomain proteins.[178–180] Domain duplications and recombination are thought to have occurred extensively in nature. Protein structure classifications are consequently domain based. Automatic recognition of domains in multidomain proteins can be difficult, although many promising approaches have been developed (see the section on protein domain above). These methods do not always agree with their domain assignments, which in turn leads to discrepancies between the existing protein structure classifications.[175,176]

The three major protein structure classifications are SCOP, CATH, and DDD. SCOP is derived manually and is recognized as a valuable resource of detailed evolutionary information. CATH provides useful geometric information. It also introduces the concept of "architecture," which reveals broad features of the protein structure space. CATH relies on partial automation and as such is subject to inaccuracies introduced by fixed thresholds. The DDD is a fully automatic classification continually updated. It is not as popular as SCOP and CATH, probably because its automatic levels are not as intuitive and require more input from the users to be interpreted.

Protein structure classifications need to be linked with the other genome databases under constructions. Currently, SCOP, CATH, and DDD are valuable resources used mostly for benchmarking of methods and for structural studies. Their impact on biology will be far greater when they are integrated with sequence and function information to present a cohesive picture of the different protein spaces.

ACKNOWLEDGMENTS

Support from the National Science Foundation (Grant CCR-00-86013) and the National Institute of Health (GM 63817) is acknowledged.

APPENDIX: HIERARCHICAL CLUSTERING

The aim of clustering is to group a collection of objects (or observations) into subsets of "clusters," such that those objects within each cluster are more similar to one another than objects assigned to other clusters. Two main elements exist in any clustering technique: the definition of similarity or dissimilarity between objects, and the algorithm that partitions the data into clusters. Here it is assumed that the similarity is known and encoded into a distance ***d*** between the objects. There are two major types of algorithm for portioning objects: k-means clustering and hierarchical clustering.

In hierarchical clustering, the data are regrouped into clusters through a series of partitioning events. Each partition can run from a single cluster containing all n objects to n clusters each containing a single object. Hierarchical clustering techniques are subdivided into two groups: *agglomerative* methods that fuse the objects into groups and *divisive* methods that separate the objects successively into finer groupings. Here the focus is on agglomerative methods, because they are used for generating protein structure classifications.

An agglomerative hierarchical clustering technique involves creating a series of partitions of the n data, P_n, P_{n-1},..., P_1, such that P_n consists of n clusters each containing a single object, and P_1 consists of a single group containing all n objects. At each stage, the procedure joins together the two nearest clusters. Differences between methods are from different ways of defining the distance between clusters. The four main agglomerative hierarchical clustering techniques are as follows:

- ***Single linkage clustering***: The distance between two clusters A and B is defined as the distance between the closest pair of objects, where only pairs consisting of one object from each cluster are considered:

$$D(A,B) = \min\{d(a,b),\ (a,b) \in A \times B\} \qquad [A.1]$$

- ***Complete linkage clustering***: The distance between two clusters A and B is defined as the distance between the most distant pair of objects, one from each cluster:

$$D(A,B) = \max\{d(a,b),\ (a,b) \in A \times B\} \qquad [A.2]$$

- ***Average linkage clustering***: The distance between two clusters A and B is defined as the average of distances between all pairs of objects, where each pair is composed of one object from each cluster:

$$D(A,B) = \frac{\sum_{a \in A} \sum_{b \in B} d(a,b)}{N_A N_B} \qquad [A.3]$$

where N_A and N_B are the sizes of A and B, respectively.

- ***Average group linkage***: The distance between two clusters A and B is defined as the average of distances between all pairs of objects included in the union of A and B:

$$D(A,B) = Average\left\{ d(a,b),\ (a,b) \in (A \cup B)^2 \right\} \qquad [A.4]$$

There is no answer to the question about which of these techniques performs best. Clustering is an exploratory data analysis procedure; the choice of which technique to be used for clustering often comes from a very good understanding of the objects to be clustered. A tutorial on clustering methods used in computational chemistry has appeared in this series and should be consulted.[181]

REFERENCES

1. J. Monod, *Le Hasard Et La Nécessité*, Seuil, Paris, France, 1973.
2. A. Bernal, U. Ear, and N. Kyrpides, *Nucl. Acids Res.*, **29**, 126 (2001). Genomes Online Database (Gold): A Monitor of Genome Projects World-Wide.
3. F. C. Bernstein, T. F. Koetzle, G. William, D. J. Meyer, M. D. Brice, and J. R. Rodgers, *J. Mol. Biol.*, **112**, 535 (1977). The Protein Databank: A Computer-Based Archival File for Macromolecular Structures.
4. H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, and H. Weissig, *Nucl. Acids Res.*, **28**, 235 (2000). The Protein Data Bank.
5. W. Gilbert, *Nature* (*London*), **349**, 99 (1991). Towards a Paradigm Shift in Biology.
6. A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, *J. Mol. Biol.*, **247**, 536 (1995). SCOP: A Structural Classification of Proteins Database for the Investigation of Sequences and Structures.
7. C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton, *Structure*, **5**, 1093 (1997). CATH: A Hierarchic Classification of Protein Domain Structures.
8. L. Holm and C. Sander, *J. Mol. Biol.*, **233**, 123 (1993). Protein Structure Comparison by Alignment of Distance Matrices.
9. G. E. Schulz and R. H. Schirmer, *Principles of Protein Structure*, Springer-Verlag, New York, 1979.
10. C. R. Cantor and P. R. Schimmel, *Biophysical Chemistry: The Conformation of Biological Macromolecules*, W. H. Freeman Company, New York, 1980.
11. C. Branden and J. Tooze, *Introduction to Protein Structure*, Garland Publishing, New York, 1991.
12. T. E. Creighton, *Proteins*, W. H. Freeman & Co., New York, 1993.
13. W. R. Taylor, A. C. W. May, N. P. Brown, and A. Aszodi, *Reports Prog. Phys.*, **64**, 517 (2001). Protein Structure: Geometry, Topology and Classification.
14. R. B. Corey and L. Pauling, *Rev. Sci. Instr.*, **24**, 621 (1953). Molecular Models of Amino Acids, Peptides and Proteins.
15. W. L. Koltun, *Biopolymers*, **3**, 665 (1965). Precision Space-Filling Atomic Models.
16. J. Kendrew, R. Dickerson, B. Strandberg, R. Hart, D. Davies, and D. Philips, *Nature* (*London*), **185**, 422 (1960). Structure of Myoglobin: A Three Dimensional Fourier Synthesis at 2 Angstrom Resolution.
17. P. J. Kraulis, *J. Appl. Cryst.*, **24**, 946 (1991). Molscript: A Program to Produce Both Detailed and Schematic Plots of Protein Structures.

18. W. Humphrey, A. Dalke, and K. Schulten, *J. Molec. Graphics*, **14**, 33 (1996). VMD - Visual Molecular Dynamics.
19. K. C. Timberlake, *General, Organic, and Biological Chemistry: Structures of Life*, Benjamin Cummings, San Francisco, CA, 2004.
20. G. M. Crippen, *J. Mol. Biol.*, **126**, 315 (1978). Tree Structural Organization of Proteins.
21. A. V. Efimov, *FEBS Lett.*, **224**, 372 (1987). Pseudo-Homology of Protein Standard Structures Formed by 2 Consecutive Beta-Strands.
22. A. V. Efimov, *FEBS Lett.*, **284**, 288 (1991). Structure of Coiled Beta-Beta Hairpins and Beta-Beta-Corners.
23. A. V. Efimov, *Protein Eng.*, **4**, 245 (1991). Structure of Alpha-Alpha Hairpins with Short Connections.
24. A. V. Efimov, *Prog. Biophys. Mol. Biol.*, **60**, 201 (1993). Standard Structures in Proteins.
25. C. Brooks, M. Karplus, and M. Pettitt, *Adv. Chem. Phys.*, **71**, 1 (1988). Proteins: A Theoretical Perspective of Dynamics, Structure and Thermodynamics.
26. C. B. Anfinsen, *Science*, **181**, 223 (1973). Principles That Govern Protein Folding.
27. P. Koehl and M. Levitt, *Nature Struct. Biol.*, **6**, 108 (1999). A Brighter Future for Protein Structure Prediction.
28. D. Baker and A. Šali, *Science*, **294**, 93 (2001). Protein Structure Prediction and Structural Genomics.
29. J.-E. Shea, M. R. Friedel, and A. Baumketner in *Rev. Comput. Chem.*, K. B. Lipkowitz, T. R. Cundari, and V. Gillet, Eds., Wiley, New York, 2006, Vol. **22**, pp. 169–228. Predicting Protein Folding.
30. M. Perutz, M. Rossman, A. Cullis, G. Muirhead, G. Will, and A. North, *Nature* (*London*), **185**, 416 (1960). Structure of Haemoglobin: A Three-Dimensional Fourier Synthesis at 5.5 Angstrom Resolution, Obtained by X-Ray Analysis.
31. M. Levitt and C. Chothia, *Nature* (*London*), **261**, 552 (1976). Structural Patterns in Globular Proteins.
32. A. M. Lesk and C. Chothia, *J. Mol. Biol.*, **136**, 225 (1980). How Different Amino-Acid Sequences Determine Similar Protein Structures: The Structure and Evolutionary Dynamics of the Globins.
33. C. Chothia and J. Janin, *Proc. Natl. Acad. Sci. (USA)*, **78**, 4146 (1981). Relative Orientation of Close Packed Beta Pleated Sheets in Proteins.
34. F. E. Cohen, M. J. E. Sternberg, and W. R. Taylor, *J. Mol. Biol.*, **148**, 253 (1981). Analysis of the Tertiary Structure of Protein Beta Sheet Sandwiches.
35. C. Chothia and J. Janin, *Biochemistry*, **21**, 3955 (1982). Orthogonal Packing of Beta Pleated Sheets in Proteins.
36. F. E. Cohen, M. J. E. Sternberg, and W. R. Taylor, *J. Mol. Biol.*, **156**, 821 (1982). Analysis and Prediction of the Packing of Alpha Helices against a Beta Sheet in the Tertiary Structure of Globular Proteins.
37. K. C. Chou, *Proteins: Struct. Func. Genet.*, **21**, 319 (1995). A Novel Approach to Predicting Protein Structural Classes in a (20-1)-D Amino Acid Composition Space.
38. K. C. Chou and C. T. Zhang, *Critical Rev. Biochem. Molec. Biol.*, **30**, 275 (1995). Prediction of Protein Structural Classes.
39. I. Bahar, A. R. Atilgan, R. L. Jernigan, and B. Erman, *Proteins: Struct. Func. Genet.*, **29**, 172 (1997). Understanding the Recognition of Protein Structural Classes by Amino Acid Composition.
40. W. M. Liu and K. C. Chou, *J. Protein Chem.*, **17**, 209 (1998). Prediction of Protein Structural Classes by Modified Mahalanobis Discriminant Algorithm.
41. K. C. Chou, W. M. Liu, G. M. Maggiora, and C. T. Zhang, *Proteins: Struct. Func. Genet.*, **31**, 97 (1998). Prediction and Classification of Domain Structural Classes.
42. Y. D. Cai, Y. X. Li, and K. C. Chou, *Biochim. Biophys. Acta*, **1476**, 1 (2000). Using Neural Networks for Prediction of Domain Structural Classes.

43. G. P. Zhou and N. Assa-Munt, *Proteins: Struct. Func. Genet.*, **44**, 57 (2001). Some Insights into Protein Structural Class Prediction.
44. R. Y. Luo, Z. P. Feng, and J. K. Liu, *Eur. J. Biochem.*, **269**, 4219 (2002). Prediction of Protein Structural Class by Amino Acid and Polypeptide Composition.
45. E. G. Hutchinson and J. M. Thornton, *Protein Eng.*, **6**, 233 (1993). The Greek Key Motif: Extraction, Classification and Analysis.
46. J. S. Richardson, *Nature* (*London*), **268**, 495 (1977). β-Sheet Topology and the Relatedness of Proteins.
47. D. W. Banner, A. C. Bloomer, G. A. Petsko, D. C. Phillips, C. I. Pogson, I. A. Wilson, P. H. Corran, A. J. Furth, J. D. Milman, R. E. Offord, J. D. Priddle, and S. G. Waley, *Nature* (*London*), **255**, 609 (1975). Structure of Chicken Muscle Triose Phosphate Isomerase Determined Crystallographically at 2.5 Å Resolution Using Amino-Acid Sequence Data.
48. A. G. Murzin, A. M. Lesk, and C. Chothia, *J. Mol. Biol.*, **236**, 1369 (1994). Principles Determining the Structure of Beta-Sheet Barrels in Proteins. 1. A Theoretical-Analysis.
49. A. G. Murzin, A. M. Lesk, and C. Chothia, *J. Mol. Biol.*, **236**, 1382 (1994). Principles Determining the Structure of Beta-Sheet Barrels in Proteins. 2. The Observed Structures.
50. G. Pujadas and J. Palau, *Biologia* (*Bratislava*), **54**, 231 (1999). Tim Barrel Fold: Structural, Functional and Evolutionary Characteristics in Natural and Designed Molecules.
51. R. K. Wierenga, *FEBS Lett.*, **492**, 193 (2001). The Tim-Barrel Fold: A Versatile Framework for Efficient Enzymes.
52. N. Nagano, C. A. Orengo, and J. M. Thornton, *J. Mol. Biol.*, **321**, 741 (2002). One Fold with Many Functions: The Evolutionary Relationships between Tim Barrel Families Based on Their Sequences, Structures and Functions.
53. M. C. Vega, E. Lorentzen, A. Linden, and M. Wilmanns, *Curr. Opin. Chem. Biol.*, **7**, 694 (2003). Evolutionary Markers in the (Beta/Alpha)8-Barrel Fold.
54. J. A. Gerlt and F. M. Raushel, *Curr. Opin. Chem. Biol.*, **7**, 252 (2003). Evolution of Function in (Beta/Alpha)8-Barrel Enzymes.
55. E. L. Wise and I. Rayment, *Acc. Chem. Res.*, **37**, 149 (2004). Understanding the Importance of Protein Structure to Nature's Routes for Divergent Evolution in Tim Barrel Enzymes.
56. G. D. Rose, *J. Mol. Biol.*, **134**, 447 (1979). Hierarchic Organization of Domains in Globular Proteins.
57. J. S. Richardson, *Adv. Protein Chem.*, **34**, 167 (1981). The Anatomy and Taxonomy of Protein Structure.
58. J. Janin and C. Chothia, *Meth. Enzymol.*, **115**, 420 (1985). Domains in Proteins - Definitions, Location, and Structural Principles.
59. C. P. Ponting and R. B. Russell, in *Annu. Rev. Biophys. Biomol. Struct.*, R. M. Stroud, W. K. Olson, and M. P. Sheetz, Eds., Annual Reviews, Palo Alto, CA, 2002, Vol. **31**, pp. 45–71. The Natural History of Protein Domains.
60. S. Veretnik, P. E. Bourne, N. N. Alexandrov, and I. N. Shindyalov, *J. Mol. Biol.*, **339**, 647 (2004). Toward Consistent Assignment of Structural Domains in Proteins.
61. R. A. Laskowski, E. G. Hutchinson, A. D. Michie, A. C. Wallace, M. L. Jones, and J. M. Thornton, *Trends Biochem. Sci.*, **22**, 488 (1997). PDBSum: A Web-Based Database of Summaries and Analyses of All PDB Structures.
62. R. A. Laskowski, *Nucl. Acids Res.*, **29**, 221 (2001). PDBSum: Summaries and Analyses of PDB Structures.
63. W. Kabsch and C. Sander, *Biopolymers*, **22**, 2577 (1983). Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen Bonded and Geometrical Features.
64. G. Wang and R. L. Dunbrack, *Bioinformatics* (*Oxford*), **19**, 1589 (2003). Pisces: A Protein Sequence Culling Server.
65. S. E. Brenner, P. Koehl, and M. Levitt, *Nucl. Acids Res.*, **28**, 254 (2000). The Astral Compendium for Protein Structure and Sequence Analysis.

66. J. M. Chandonia, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. E. Brenner, *Nucl. Acids Res.*, **30**, 260 (2002). Astral Compendium Enhancements.

67. J. M. Chandonia, G. Hon, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. E. Brenner, *Nucl. Acids Res.*, **32**, D189 (2004). The Astral Compendium in 2004.

68. M. G. Rossman and A. Liljas, *J. Mol. Biol.*, **85**, 177 (1974). Recognition of Structural Domains in Globular Proteins.

69. M. H. Zehfus and G. D. Rose, *Biochemistry*, **25**, 5759 (1986). Compact Units in Proteins.

70. S. A. Islam, J. C. Luo, and M. J. E. Sternberg, *Protein Eng.*, **8**, 513 (1995). Identification and Analysis of Domains in Proteins.

71. A. S. Siddiqui and G. J. Barton, *Protein Sci.*, **4**, 872 (1995). Continuous and Discontinuous Domains: An Algorithm for the Automatic Generation of Reliable Protein Domain Definitions.

72. N. Alexandrov and I. Shindyalov, *Bioinformatics* (*Oxford*), **19**, 429 (2003). PDB: Protein Domain Parser.

73. S. J. Wodak and J. Janin, *Biochemistry*, **20**, 6544 (1981). Location of Structural Domains in Proteins.

74. A. A. Rashin, *Nature* (*London*), **291**, 85 (1981). Location of Domains in Globular Proteins.

75. N. Go, *Proc. Natl. Acad. Sci.* (*USA*), **80**, 1964 (1983). Modular Structural Units, Exon and Function in Chicken Lysozyme.

76. M. H. Zehfus, *Protein Eng.*, **7**, 335 (1994). Binary Discontinuous Compact Protein Domains.

77. L. Holm and C. Sander, *Proteins: Struct. Func. Genet.*, **19**, 256 (1994). Parser for Protein Folding Units.

78. Y. Xu, D. Xu, and H. N. Gambow, *Bioinformatics* (*Oxford*), **16**, 1091 (2000). Protein Domain Decomposition Using a Graph-Theoretic Approach.

79. J. T. Guo, D. Xu, D. Kim, and Y. Xu, *Nucl. Acids Res.*, **31**, 944 (2003). Improving the Performance of Domainparser for Structural Domain Partition Using Neural Network.

80. W. R. Taylor, *Protein Eng.*, **12**, 203 (1999). Protein Structural Domain Identification.

81. M. B. Swindells, *Protein Sci.*, **4**, 93 (1995). A Procedure for the Automatic Determination of Hydrophobic Cores in Protein Structures.

82. M. B. Swindells, *Protein Sci.*, **4**, 103 (1995). A Procedure for Detecting Structural Domains in Proteins.

83. A. S. Siddiqui and G. J. Barton, *Protein Sci.*, **4**, 872 (1995). Continuous and Discontinuous Domains: An Algorithm for the Automatic-Generation of Reliable Protein Domain Definitions.

84. C. Guerra and S. Istrail, *Mathematical Methods for Protein Structure Analysis and Design, Advanced Lectures*, Springer, Berlin, Germany, 2003.

85. C. Chen and Q. Li, *Acta Cryst. A*, **A60**, 201 (2004). A Strict Solution for the Optimal Superposition of Protein Structures.

86. B. Sabata and J. K. Aggarwal, *Comput. Vis. Graph. Image Proc: Image Understanding*, **54**, 309 (1991). Estimation of Motion from a Pair of Range Images: A Review.

87. C. Ferrari and C. Guerra, in *Mathematical Methods for Protein Structure Analysis and Design*, C. Guerra and S. Istrail, Eds., Springer, Berlin, Germany, 2003, pp. 57–82. Geometric Methods for Protein Structure Comparison.

88. D. W. Eggert, A. Lorusso, and R. B. Fisher, *Mach. Vis. and Applic.*, **9**, 272 (1997). Estimating 3D Rigid Body Transformations: A Comparison of Four Major Algorithms.

89. G. H. Golub and C. F. V. Loan, *Matrix Computation*, John Hopkins University Press, Baltimore, MD, 1996.

90. W. Kabsch, *Acta Cryst. A*, **32**, 922 (1976). Solution for Best Rotation to Relate 2 Sets of Vectors.

91. W. Kabsch, *Acta Cryst. A*, **34**, 827 (1978). Discussion of Solution for Best Rotation to Relate 2 Sets of Vectors.
92. A. D. McLachlan, *J. Mol. Biol.*, **128**, 49 (1979). Gene Duplications in the Structural Evolution of Chymotrypsin.
93. K. S. Arun, T. S. Huang, and S. D. Blostein, *IEEE Trans. Pattern Anal. & Machine Intel.*, **9**, 698 (1987). Least-Sqaure Fitting of Two 3D Point Sets.
94. P. H. Schonemann, *Psychometrica*, **31**, 1 (1966). A Generalized Solution of the Orthogonal Procrustes Problem.
95. B. Horn, H. Hilden, and S. Negahdaripour, *J. Opt. Soc. Am.*, **5**, 1127 (1988). Closed-Form Solution of Absolute Orientation Using Orthonormal Matrices.
96. B. Horn, *J. Opt. Soc. Am.*, **4**, 629 (1987). Closed-Form Solution of Absolute Orientation Using Unit Quaternions.
97. M. W. Walker, L. Shao, and R. A. Voltz, *CVGIP: Image Understanding*, **54**, 358 (1991). Estimating 3D Location Parameters Using Dual Number Quaternions.
98. E. A. Coutsias, C. Seok, and K. A. Dill, *J. Comput. Chem*, **25**, 1849 (2004). Using Quaternions to Calculate RMSD.
99. S. Umeyama, *IEEE Trans. Pattern Anal. & Machine Intel.*, **13**, 376 (1991). Least-Squares Estimation of Transformation Parameters between 2-Point Patterns.
100. P. Koehl, *Curr. Opin. Struct. Biol.*, **11**, 348 (2001). Protein Structure Similarities.
101. W. R. Taylor and C. A. Orengo, *J. Mol. Biol.*, **208**, 1 (1989). Protein Structure Alignment.
102. W. R. Taylor, *Protein Sci.*, **8**, 654 (1999). Protein Structure Comparison Using Iterated Double Dynamic Programming.
103. K. Nishikawa and T. Ooi, *J. Theor. Biol.*, **43**, 351 (1974). Comparison of Homologous Tertiary Structures of Proteins.
104. L. Holm, C. Ouzounis, C. Sander, G. Tuparev, and G. Vriend, *Protein Sci.*, **1**, 1691 (1992). A Database of Protein Structure Families with Common Folding Motifs.
105. L. Holm and C. Sander, *Nature* (*London*), **361**, 309 (1993). Globin Fold in a Bacterial Toxin.
106. G. Vriend and C. Sander, *Proteins: Struct. Func. Genet.*, **11**, 52 (1991). Detection of Common 3-Dimensional Substructures in Proteins.
107. N. N. Alexandrov, K. Takahashi, and N. Go, *J. Mol. Biol.*, **225**, 5 (1992). Common Spatial Arrangements of Backbone Fragments in Homologous and Nonhomologous Proteins.
108. R. Nussinov and H. J. Wolfson, *Proc. Natl. Acad. Sci.* (*USA*), **88**, 10495 (1991). Efficient Detection of 3-Dimensional Structural Motifs in Biological Macromolecules by Computer Vision Techniques.
109. R. Nussinov, D. Fischer, and H. Wolfson, *FASEB J.*, **6**, A349 (1992). A Computer Vision Based 3-Dimensional Approach for the Comparison of Protein Structures.
110. D. Fischer, O. Bachar, R. Nussinov, and H. Wolfson, *J. Biomol. Struct. Dyn.*, **9**, 769 (1992). An Efficient Automated Computer Vision Based Technique for Detection of 3-Dimensional Structural Motifs in Proteins.
111. D. Fischer, H. Wolfson, and R. Nussinov, *J. Biomol. Struct. Dyn.*, **11**, 367 (1993). Spatial, Sequence-Order-Independent Structural Comparison of Alpha/Beta Proteins: Evolutionary Implications.
112. D. Fischer, H. Wolfson, S. L. Lin, and R. Nussinov, *Protein Sci.*, **3**, 769 (1994). 3-Dimensional, Sequence Order-Independent Structural Comparison of a Serine-Protease against the Crystallographic Database Reveals Active-Site Similarities - Potential Implications to Evolution and to Protein-Folding.
113. H. J. Wolfson and I. Rigoutsos, *IEEE Comput. Sci. & Eng.*, **4**, 10 (1997). Geometric Hashing: An Overview.
114. P. J. Artymiuk, A. R. Poirrette, H. M. Grindley, D. W. Rice, and P. Willett, *J. Mol. Biol.*, **243**, 327 (1994). A Graph-Theoretic Approach to the Identification of 3-Dimensional Patterns of Amino-Acid Side-Chains in Protein Structures.

115. P. J. Artymiuk, A. R. Poirrette, D. W. Rice, and P. Willett, *Topics Curr. Chem.*, **174**, 73 (1995). The Use of Graph-Theoretical Methods for the Comparison of the Structures of Biological Macromolecules.

116. E. J. Gardiner, P. J. Artymiuk, and P. Willett, *J. Mol. Graph. & Modelling*, **15**, 245 (1997). Clique-Detection Algorithms for Matching Three-Dimensional Molecular Structures.

117. T. D. Wu, S. C. Schmidler, T. Hastie, and D. L. Brutlag, *J. Comput. Biol.*, **5**, 585 (1998). Regression Analysis of Multiple Protein Structures.

118. S. Subbiah, D. V. Laurents, and M. Levitt, *Curr. Biol.*, **3**, 141 (1993). Structural Similarity of DNA-Binding Domains of Bacteriophage Repressors and the Globin Core.

119. M. Gerstein and M. Levitt, *Protein Sci.*, **7**, 445 (1998). Comprehensive Assessment of Automatic Structural Alignment against a Manual Standard; the SCOP Classification of Proteins.

120. A. S. Yang and B. Honig, *J. Mol. Biol.*, **301**, 665 (2000). An Integrated Approach to the Analysis and Modeling of Protein Sequences and Structures. I. Protein Structural Alignment and a Quantitative Measure for Protein Structural Distance.

121. J. D. Szustakowski and Z. P. Weng, *Proteins: Struct. Func. Genet.*, **38**, 428 (2000). Protein Structure Alignment Using a Genetic Algorithm.

122. I. N. Shindyalov and P. E. Bourne, *Protein Eng.*, **11**, 739 (1998). Protein Structure Alignment by Incremental Combinatorial Extension (CE) of the Optimal Path.

123. M. E. Ochagavia and S. J. Wodak, *Proteins: Struct. Func. Bioinfo.*, **55**, 436 (2004). Progressive Combinatorial Algorithm for Multiple Structural Alignments: Application to Distantly Related Proteins.

124. R. Blankenbecler, M. Ohlson, C. Peterson, and M. Ringler, *Proc. Natl. Acad. Sci. (USA)*, **100**, 11936 (2003). Matching Protein Structures with Fuzzy Alignments.

125. L. Chen, T. Zhou, and Y. Tang, *Bioinformatics* (*Oxford*), **21**, 51 (2005). Protein Structure Alignment by Deterministic Annealing.

126. B. W. Matthews and M. G. Rossman, *Meth. Enzymol.*, **115**, 397 (1985). Comparison of Protein Structures.

127. M. L. Sierk and W. R. Pearson, *Protein Sci.*, **13**, 773 (2004). Sensitivity and Selectivity in Protein Structure Comparison.

128. M. Novotny, D. Madsen, and G. J. Kleywegt, *Proteins: Struct. Func. Genet.*, **54**, 260 (2004). Evaluation of Protein Fold Comparison Servers.

129. R. Kolodny, P. Koehl, and M. Levitt, *J. Mol. Biol.*, **346**, 1173 (2005). Comprehensive Evaluation of Structural Alignment Method: Scoring by Geometric Match Measures.

130. L. Holm and C. Sander, *Trends Biochem. Sci.*, **20**, 478 (1995). DALI - a Network Tool for Protein-Structure Comparison.

131. M. G. Rossmann and P. Argos, *J. Mol. Biol.*, **105**, 75 (1976). Exploring Structural Homology of Proteins.

132. R. B. Russell and G. J. Barton, *Proteins: Struct. Func. Genet.*, **14**, 309 (1992). Multiple Protein Sequence Alignment from Tertiary Structure Comparison Assignment of Global and Residue Confidence Levels.

133. M. Suyama, Y. Matsuo, and K. Nishikawa, *J. Mol. Evol.*, **44**, S163 (1997). Comparison of Protein Structures Using 3D Profile Alignment.

134. J. U. Bowie, R. Lüthy, and D. Eisenberg, *Science*, **253**, 164 (1991). A Method to Identify Protein Sequences That Fold into a Known Three-Dimensional Structure.

135. J. Jung and B. Lee, *Protein Eng.*, **13**, 535 (2000). Protein Structure Alignment Using Environmental Profiles.

136. T. Kawabata and K. Nishikawa, *Proteins: Struct. Func. Genet.*, **41**, 108 (2000). Protein Structure Comparison Using the Markov Transition Model of Evolution.

137. I. D. Kuntz, G. M. Crippen, P. A. Kollman, and D. Kimelman, *J. Mol. Biol.*, **106**, 983 (1976). Calculation of Protein Tertiary Structure.

138. G. M. Crippen and T. F. Havel, *Distance Geometry and Molecular Conformation*, Research Studies, Somerset, United Kingdom, 1988.

139. R. Kolodny and N. Linial, *Proc. Natl. Acad. Sci. (USA)*, **101**, 12201 (2004). Approximate Protein Structural Alignment in Polynomial Time.

140. K. Kaindl and B. Steipe, *Acta Cryst. A*, **53**, 809 (1997). Metric Properties of the Root-Mean-Square Deviation of Vector Sets.

141. K. Mizuguchi and N. Go, *Curr. Opin. Struct. Biol.*, **5**, 377 (1995). Seeking Significance in 3-Dimensional Protein-Structure Comparisons.

142. A. M. Lesk, *Proteins: Struct. Func. Genet.*, **33**, 320 (1998). Extraction of Geometrically Similar Substructures: Least-Squares and Chebyshev Fitting and the Difference Distance Matrix.

143. A. C. W. May, *Proteins: Struct. Func. Genet.*, **37**, 20 (1999). Toward More Meaningful Hierarchical Classification of Protein Three-Dimensional Structures.

144. C. A. Orengo, M. B. Swindells, A. D. Michie, M. J. Zvelebil, P. C. Driscoll, M. D. Waterfield, and J. M. Thornton, *Protein Sci.*, **4**, 1977 (1995). Structure Similarity between the Pleckstrin Homology Domain and Verotoxin: The Problem of Measuring and Evaluating Structural Similarity.

145. Z. K. Feng and M. J. Sippl, *Folding & Design*, **1**, 123 (1996). Optimum Superimposition of Protein Structures: Ambiguities and Implications.

146. A. Godzik, *Protein Sci.*, **5**, 1325 (1996). The Structural Alignment between Two Proteins: Is There a Unique Answer?

147. T. J. P. Hubbard, *Proteins: Struct. Func. Genet.*, **Suppl 3**, 15 (1999). RMS/Coverage Graphs: A Qualitative Method for Comparing Three-Dimensional Protein Structure Predictions.

148. G. A. Arteca, in *Rev. Comput. Chem.*, K. B. Lipkowitz and D. B. Boyd, Eds., VCH Publishers, New York, 1996, Vol. **9**, pp. 191–254. Molecular Shape Descriptors.

149. P. Rogen and H. Bohr, *Math. Biosci.*, **182**, 167 (2003). A New Family of Global Protein Shape Descriptors.

150. D. Barnatan, *Topology*, **34**, 423 (1995). On the Vassiliev Knot Invariants.

151. A. Stasiak and J. H. Maddocks, *Nature* (*London*), **406**, 251 (2000). Mathematics - Best Packing in Proteins and DNA.

152. K. A. Hoffman, R. S. Manning, and J. H. Maddocks, *Biopolymers*, **70**, 145 (2003). Link, Twist, Energy, and the Stability of DNA Minicircles.

153. M. Levitt, *J. Mol. Biol.*, **170**, 723 (1983). Protein Folding by Restrained Energy Minimization and Molecular Dynamics.

154. G. A. Arteca, *Biopolymers*, **33**, 1829 (1993). Overcrossing Spectra of Protein Backbones - Characterization of 3-Dimensional Molecular Shape and Global Structural Homologies.

155. G. A. Arteca, *Phys. Rev. E*, **49**, 2417 (1994). Scaling Behavior of Some Molecular Shape Descriptors of Polymer Chains and Protein Backbones.

156. G. A. Arteca, *Phys. Rev. E*, **51**, 2600 (1995). Scaling Regimes Self-Entanglements in Very Compact Proteins.

157. G. A. Arteca and O. Tapia, *J. Chem. Inf. Comput. Sci.*, **39**, 642 (1999). Characterization of Fold Diversity among Proteins with the Same Number of Amino Acid Residues.

158. C. T. Reimann, G. A. Arteca, and O. Tapia, *Phys. Chem. Chem. Phys.*, **4**, 4058 (2002). Proteins in Vacuo. A Connection between Mean Overcrossing Number and Orientationally-Averaged Collision Cross Section.

159. P. Rogen and B. Fain, *Proc. Natl. Acad. Sci. (USA)*, **100**, 119 (2003). Automatic Classification of Protein Structure by Using Gauss Integrals.

160. J. H. White, *Am. J. Math.*, **91**, 693 (1969). Self-Linking and Gauss-Integral in Higher Dimensions.

161. O. Gonzalez and J. H. Maddocks, *Proc. Natl. Acad. Sci. (USA)*, **96**, 4769 (1999). Global Curvature, Thickness, and the Ideal Shapes of Knots.

162. A. Maritan, C. Micheletti, A. Trovato, and J. R. Banavar, *Nature* (*London*), **406**, 287 (2000). Optimal Shapes of Compact Strings.

163. J. R. Banavar, A. Maritan, C. Micheletti, and A. Trovato, *Proteins: Struct. Func. Genet.*, **47**, 315 (2002). Geometry and Physics of Proteins.

164. J. R. Banavar, A. Maritan, and F. Seno, *Proteins: Struct. Func. Genet.*, **49**, 246 (2002). Anisotropic Effective Interactions in a Coarse-Grained Tube Picture of Proteins.

165. J. R. Banavar and A. Maritan, *Rev. Modern Phys.*, **75**, 23 (2003). Colloquium: Geometrical Approach to Protein Folding: A Tube Picture.

166. J. R. Banavar, A. Flammini, D. Marenduzzo, A. Maritan, and A. Trovato, *J. Phys: Cond. Matter*, **15**, S1787 (2003). Tubes near the Edge of Compactness and Folded Protein Structures.

167. T. X. Hoang, A. Trovato, F. Seno, J. R. Banavar, and A. Maritan, *Proc. Natl. Acad. Sci.* (*USA*), **101**, 7960 (2004). Geometry and Symmetry Presculpt the Free-Energy Landscape of Proteins.

168. C. A. Ouzounis, R. M. R. Coulson, A. J. Enright, V. Kunin, and J. B. Pereira-Leal, *Nature Rev. Genet.*, **4**, 508 (2003). Classification Schemes for Protein Structure and Function.

169. A. D. Michie, C. A. Orengo, and J. M. Thornton, *J. Mol. Biol.*, **262**, 168 (1996). Analysis of Domain Structural Class Using an Automated Class Assignment Protocol.

170. S. Jones, M. Stewart, A. Michie, M. B. Swindells, C. Orengo, and J. M. Thornton, *Protein Sci.*, **7**, 233 (1998). Domain Assignment for Protein Structures Using a Consensus Approach: Characterization and Analysis.

171. L. Holm and C. Sander, *Science*, **273**, 595 (1996). Mapping the Protein Universe.

172. S. Dietmann and L. Holm, *Nature Struct. Biol.*, **8**, 953 (2001). Identification of Homology in Protein Structure Classification.

173. L. Holm and C. Sander, *Nucl. Acids Res.*, **26**, 316 (1998). Touring Protein Fold Space with DALI/FSSP.

174. L. Holm and C. Sander, *Nucl. Acids Res.*, **22**, 3600 (1994). The FSSP Database of Structurally Aligned Protein Fold Families.

175. C. Hadley and D. T. Jones, *Structure*, **7**, 1099 (1999). A Systematic Comparison of Protein Structure Classifications: SCOP, CATH and FSSP.

176. R. Day, D. A. C. Beck, R. S. Armen, and V. Daggett, *Protein Sci.*, **12**, 2150 (2003). A Consensus View of Fold Space: Combining SCOP, CATH, and the DALI Domain Dictionary.

177. A. Harrison, F. Pearl, R. Mott, J. Thornton, and C. Orengo, *J. Mol. Biol.*, **323**, 909 (2002). Quantifying the Similarities within Fold Space.

178. G. Apic, J. Gough, and S. Teichmann, *J. Mol. Biol.*, **310**, 311 (2001). Domain Combinations in Archaeal, Eubacterial and Eukaryotic Proteomes.

179. S. Teichmann, A. G. Murzin, and C. Chothia, *Curr. Opin. Struct. Biol.*, **11**, 354 (2001). Determination of Protein Function, Evolution and Interactions by Structural Genomics.

180. B. Rost, *Curr. Opin. Struct. Biol.*, **12**, 409 (2002). Did Evolution Leap to Create the Protein Universe?

181. G. M. Downs and J. M. Barnard, in *Rev. Comput. Chem.*, K. B. Lipkowitz, R. Larter, and T. Cundari, Eds., Wiley-VCH, New York, 2002, Vol. **18**, pp. 1–40. Clustering Methods and Their Uses in Computational Chemistry.

CHAPTER 2

Comparative Protein Modeling

Emilio Xavier Esposito,[*] Dror Tobi,[†] and Jeffry D. Madura[‡]

[*]*Molecular Modelling and Bioinformatics Studio, Department of Chemistry and Molecular Biology, North Dakota State University, Fargo, North Dakota 58105*
[†]*Department of Computational Biology, School of Medicine, University of Pittsburgh, Pittsburgh, Pennsylvania 15213*
[‡]*Department of Chemistry and Biochemistry and the Center for Computational Sciences, Duquesne University, Pittsburgh, Pennsylvania 15282*

INTRODUCTION

Comparative modeling, also known as homology modeling, is a reliable computational tool to predict the three-dimensional (3-D) structure of proteins whose structures are unknown. The relationship between the sequence and the structure of a protein is well established; every protein with a known 3-D structure may thus serve as a template that can predict the structure of a protein whose structure is not known if they share sequence similarity. The term ''homology'' refers to the evolutionary relationship between two or more proteins having the same ancestor in an evolution tree regardless of their sequence similarity (which can be very low). Therefore, the term ''comparative modeling'' is more accurate than is the term ''homology modeling'' for the purposes

Reviews in Computational Chemistry, Volume 22
edited by Kenny B. Lipkowitz, Thomas R. Cundari, and Valerie J. Gillet

of predicting 3-D shape and function as there need not be an ancestral similarity between the unknown protein and the known template.[1] Comparative modeling includes the fundamentals of homology modeling in addition to the construction of protein models from templates possessing the same or similar structure (but that may have different biological functions). Proteins from similar families often have similar functions; yet there are many instances in which proteins have similar structure but different functions.[2] This chapter describes the steps needed to construct a protein model beginning from an initial search for similar sequences and ending with an evaluation of the final model.

The first comparative (homology) modeling article was published by Browne et al.[3] who derived the 3-D structure of bovine α-lactalbumin (BCLA) based on hen's egg-white lysozyme. Using methodologies similar to those used today, the primary structures, i.e., the sequences, of the target (α-lactalbumin) and suitable templates (sperm whale myoglobin (SWM), horse hemoglobin α and β (HBA) and (HBB), and hen's egg-white lysozyme) were aligned. The best template (hen's egg-white lysozyme) was then selected based on its sequence similarity and the configuration of its disulfide bridges. The enzymes of interest (template and target) have similar tertiary structures, but they have different biological functions. Construction of the target model (α-lactalbumin) began with a mechanical wire model of lysozyme and was completed with techniques defined by Blake et al.[4] The wire lysozyme model was transformed (based on the alignment) into α-lactalbumin with a method similar to the segment matching method (SMM) of Levitt[5] that will be described later in this chapter. Identical residues between template and target were retained. Residues of the target differing from those of the template had their side chains modified with minimal conformational change. The backbone was adjusted to accommodate changes in length where deletions existed. In this historic publication, residues were removed from loop regions of the lysozyme template to preserve the structure and location of α-helices. The derived protein model had hydrophobic residues that were replaced with other hydrophobic residues[3] that were complementary with respect to spatially neighboring residues.[3] Browne et al. cautioned that their proposed 3-D structure of α-lactalbumin might not be correct. Their work foreshadowed our current practice of using an existing protein as a template for predicting another protein's tertiary structure, and they noted that this method has more potential for success than relying on just "chemical information alone."[3] The work of Browne et al.[3] is viewed by many as laying the groundwork for other comparative protein modeling methodologies. Comparative modeling involves the alignment of target and template sequences and then using the template structure as a framework for the construction of a structurally and/or functionally related protein. The concepts of locating templates, carrying out alignments, and accounting for missing amino acids (deletions) are new to novice modelers and will be described here.

The fields of comparative and homology modeling share a common background and terminology. The most common terms used by researchers in both fields refer to the protein being modeled (the target) and the protein structure(s) that construct the 3-D model [the template(s)]. In this chapter, the term "protein" includes enzymes and other biological receptor macromolecules of interest. A typical protein modeling endeavor consists of four main steps: (1) finding the templates, (2) aligning the target to the template, (3) constructing the protein model, and (4) evaluating the derived model as discussed in the review by Marti-Renom et al.[6] In this chapter, for the sake of pedagogy, steps 1 and 3 are further divided. Finding the template is partitioned into two stages: finding related sequences and 3-D structures and selecting a template. Likewise, constructing the protein model is divided into two separate steps: building the protein models and refining the model's structure. Construction of a 3-D model of a protein thus consists of six main steps: (1) Find the known sequences and 3-D structures that are related to the target protein of interest, (2) align the target and template amino acid residues, (3) select the templates and adjust the alignments, (4) construct, (5) refine, and (6) evaluate the model. Figure 1 shows the interconnectedness of these steps. The most difficult steps to carry out are 2 and 6, the alignment of the amino acid residues and the

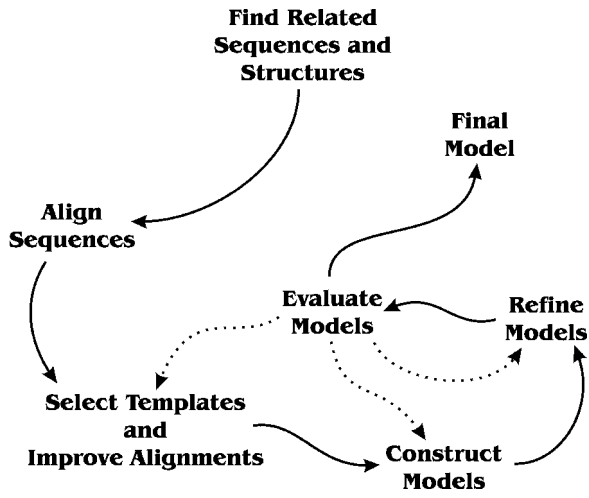


Figure 1 Flow chart of a typical comparative protein modeling study. The solid lines represent the flow of constructing a comparative protein model. The dotted lines indicate the steps where parameters (template, alignment, construction environment, or refinement method) can be modified to improve the quality of the protein model. For example, after evaluating a protein model, modelers often discuss the quality of the alignment. Instead of realigning the target sequence to the template, the alignment is "tweaked" (improved slightly) to position a gap into a loop region. After the tweaking, new protein models are constructed, refined, and again evaluated. The "new" protein model is considered correct, and the final model is ready for exploration.

evaluation of the final model. It does not imply that the other steps are trivial, however. Many aspects of comparative modeling can be automated, but building and evaluating the "best" models involve significant user intervention. The steps described here are similar in concept to those outlined in the MODELLER user manual that is available on the Internet.[7]

Comparative modeling can be considered to be one of the many endpoints for the discipline of bioinformatics where information gathered and analyzed on the relationship between amino acid sequences, distant (and not so distant) evolutionary relationships, and protein function and structure are paramount. For more information on bio-informatics, the reader should consult the books by Mount,[8] Tisdall,[9,10] Gibas and Jambeck,[11] Attwood and Parry-Smith,[12] and the two-volume set on bio-informatics and drug discovery by Lengauer et al.[13,14] The concept that proteins with similar sequences should possess similar structures is the basis for building protein models of distantly related proteins. Following the six steps depicted in Figure 1 does not ensure that a correct model will be constructed; rather it ensures only that a statistically sound model has been created. Throughout this chapter we shall present several different methods for each step of the protein modeling process (Figure 1) along with background information about each method discussed, which thus enables you to understand the processes and theory of the methods.

Three basic comparative modeling scenarios[6] can be envisioned depending on the similarity (percent identity) of the protein sequence being modeled (target) compared with the protein structure(s) being used as the reference structure (template): (1) very similar (>60% identical sequences), (2) moderately similar (between 30% and 60% identical sequences), and (3) dissimilar (<30% identical sequences). The methodologies discussed here are applicable to all categories, albeit to a varying degree.

This chapter introduces comparative modeling in the same sequence that one would use in constructing a protein model. In the next section, the steps that construct a protein model by comparative and homology modeling are outlined. The methods and applications are discussed in sufficient detail to allow a novice to become knowledgeable in rudimentary homology modeling. Several examples are presented and discussed in each step of the Anatomy section. Those researchers with a new interest in protein modeling but who lack a basic understanding of biochemistry are encouraged to read the primers on protein structure by Branden and Tooze,[2] Petsko and Ringe,[15] Creighton,[16] and Nelson et al.,[17] whereas those researchers interested in additional comparative protein modeling resources are directed to *Current Protocols in Bioinformatics*.[18,19]

ANATOMY OF A COMPARATIVE MODEL

The process of determining the best alignment and how to construct, refine, and evaluate protein models is less than a precise science and more

of an art. This review provides information about each of the logical steps involved in comparative modeling. We also provide different methods of accomplishing the same goal for each step in addition to background information about the methods implemented.

When discussing the construction of protein models with other scientists and when reading the literature, the topic of ab initio protein folding will be mentioned. The prediction of a protein's tertiary structure can be done with either comparative modeling methods (the focus of this chapter) or ab initio methods. Details of the ab initio protein cannot be discussed adequately here, so the reader is directed to reviews by Harden et al.[20] and Bonneau and Baker.[21] Also, the tutorial in this volume of *Reviews in Computational Chemistry* by Shea et al.[22] describes many of the ab initio methods for predicting protein structure from sequence information and should be consulted. Some methods used in ab initio folding are the same as those used in comparative modeling. The main difference is the use of a template by comparative modeling compared with first principles methods by ab initio protein folding to construct a protein model.

As mentioned, the construction of a protein model can be accomplished in six steps. Here we discuss each of those steps and explain several methodologies (applications) available for each. The most noticeable theme about a comparative modeling study is that some steps allow for many ways to complete the same task. For each step, only the most common methods will be discussed along with their advantages and disadvantages.

STEP 1: SEARCHING FOR RELATED SEQUENCES AND STRUCTURES

The first step toward constructing a model of the desired protein is acquiring the amino acid residue sequence (the primary structure) and then finding related sequences and solved 3-D structures of pertinent proteins. This step is where bio-informatics and computational biology/chemistry are intimately linked. Using an organism's genome, it is possible to determine the amino acid sequences of all proteins expressed by that organism. The large amount of experimental data being generated on a daily basis is great for science, but without repositories and databases, there is no efficient way to sort and search for meaningful information.

It is estimated that the amount of genomic data collected and analyzed doubles yearly.[23] Without suitable databases, it would be difficult to use the results of genome sequencing. The journal *Nucleic Acids Research*[24] publishes an annual database issue[25] with reviews covering all databases freely available (the exception being for databases that provide limited access for those who choose not to register). Those databases provide annotation-based searches with the option (links to other websites) to perform sequence similarity

searches. A problem is that it is not uncommon for a protein to have more than one accepted name; typically a protein's name changes from when it is first discovered and later in time when more is learned about its true origin and function. Additionally, some proteins have more than one domain (a well-defined structural unit of a continuous peptide chain that might have a separate function from the rest of the protein) and an annotated search has the possibility of returning these proteins as a possible match. The results obtained from an annotated database should be examined carefully to ensure that the desired protein and related proteins have been found. Chapter 1 by Koehl provides an in-depth tutorial on protein structure classification and provides related information on such databases.[26]

Expert Protein Analysis System (ExPASy)

For the scientist interested in biological research, the ExPASy website[27] is the best place to start for the retrieval of a protein's sequence and related sequences. The ExPASy website not only stores and provides access to this data, but it also provides a seamless way of moving from one area of interest to another. The ExPASy website is located at the Swiss Institute of Bioinformatics and houses four regularly updated databases focusing on proteins and proteomics, along with other protein-related databases and analysis tools. An initial search with ExPASy can lead to many other database sites, so the original search can be used as a "jumping off point" to find additional information about the system of interest. The ExPASy databases include Swiss-Prot and TrEMBL[28] (part of the Universal Protein[29] (UniProt) resource), PROSITE,[30] ENZYME,[31] and SWISS-MODEL Repository,[32] which are four areas containing a wealth of information.

Swiss-Prot and TrEMBL[28] are the initial databases one would use to find information regarding the protein's sequence. Depending on the level of refinement and information provided for each sequence, data are deposited into either Swiss-Prot or TrEMBL. Because of the large number of sequences that are now determined from genomics alone, all sequences are first deposited into TrEMBL. Once the correct nomenclature, links to relevant databases, and significant comments are added, and after unnecessary information has been removed, the sequence is added to the Swiss-Prot database. The initial search for a sequence is typically performed on the information contained in these two databases.

The PROSITE database[30] is used to determine the domain and the family of the protein sequences that, in turn, are based on biologically significant sites, patterns, and profiles.[27] This database is similar to the HOMologous STRucture Alignment Database[33,34] (HOMSTRAD) and the Protein family[35] (Pfam) database, both of which contain domain and family information for proteins. HOMSTRAD uses sequence and structure to group proteins into domains and families. Pfam classifies protein domains and families, based

on their primary structure, using information from the Swiss-Prot and TrEMBL databases. The myriad of different proteins can be grouped into specific domains and families based on their sequence and structure because it is likely they come from a common ancestor (protein). These databases exploit the conservation of tertiary structure and key amino acid residues, which makes it possible to predict the domain and family of new proteins.

The ExPASy website provides a researcher with the ability to find enzymes related to the enzyme of interest by searching with the Enzyme Committee (EC) number. The EC number is based on the recommendation of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology[31] (IUBMB). The information retrieved by the ExPASy ENZYME search includes the recommended name, alternative names (if any), catalytic activity, cofactors (if any), human genetic diseases (if any), and cross-references. The EC number consists of four segments (A.B.C.D). The first segment (A) notates the class of enzyme, the second segment (A.B) denotes the type of enzymatic reaction, the third segment (A.B.C) further defines the type of enzymatic reaction, and the fourth segment (A.B.C.D) is the classification of the enzymatic reaction. The importance of the EC number is not obvious at first, but invoking the concept that structure and function are related for proteins and enzymes makes it possible to find related enzymes based on function quickly.

The SWISS-MODEL repository[32] component of ExPASy contains comparative models of proteins without a known 3-D structure. It is comparable with the theoretical protein model section of the Protein Data Bank[36] (PDB). In contrast to the PDB, the contents of the SWISS-MODEL repository are constructed with a fully automated system. The repository is continuously updated with protein models based on new or modified sequences using new templates. The parameters used to construct the models and the resulting models are available for examination. The database contains general information about the model, the alignment, the validation report, and the modeling log. The general information provides the name of the model, the residue range, the template, the sequence identity, and the alignment *E*-values (the *E*-value score is reported by a Basic Local Alignment Search Tool[37,38] (BLAST) search that is discussed in a later section). The alignment of the target to the template is also provided, which illustrates conserved and similar residues in addition to the secondary structure of the target and template protein structures. The validation of the protein model is performed with two methods that evaluate individual amino acid residues. The first method, Atomic Non-Linear Environment Assessment[39] (ANOLEA), calculates the atomic mean force potential of the proposed protein model and the nonlocal energy profile. The second method calculates the force field energies of the individual amino acid residues with GROMOS96.[40]

The results of these tests are displayed as bar graphs, where positive values indicate misaligned or misfolded regions and negative values indicate

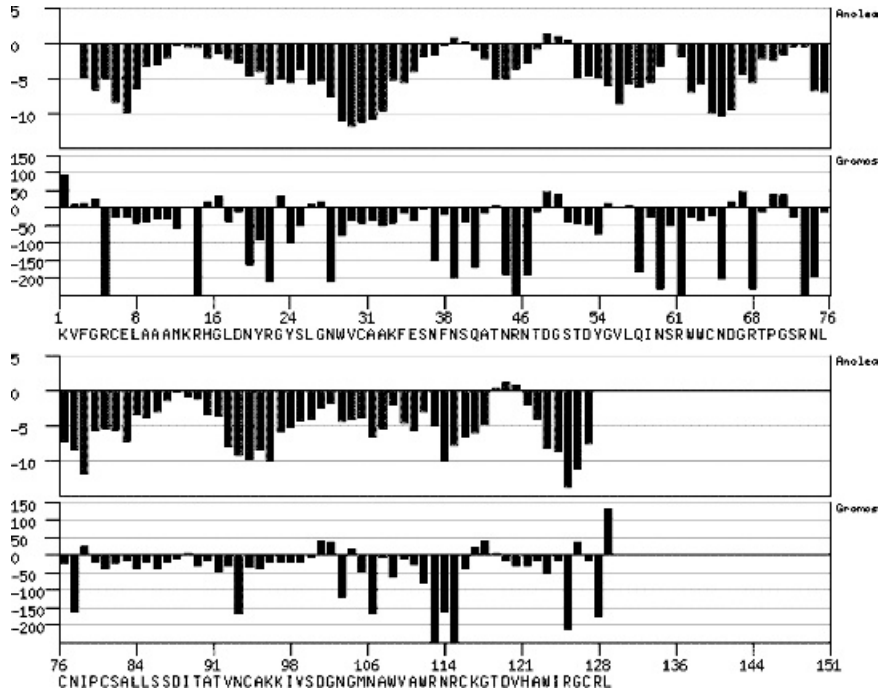


Figure 2 ANOLEA and GROMOS analysis of the California Quail Lysozyme C (P00699) protein model constructed by SWISS-MODEL.[32] For both analysis methods, negative values are preferred and indicate structurally sound models.

segments that are energetically favorable (Figure 2). The protein models are output in PDB[36] and DeepView[41] project formats. The DeepView project file gives the user the ability to modify the initial alignment to construct new models. The modeling log contains information about the template (structure information and reference), the alignment (also available is the output from the selection of the template structure), the iterative refinement of the loops, and the evaluation of the final model.

The protein models provided by the SWISS-MODEL repository can be considered to be good starting points for the construction of a 3-D model but cannot be considered to be the final structure for one main reason—the alignment. The proposed structure from SWISS-MODEL is not the result of aligning sequences from many similar proteins to aid in the discovery of conserved amino acid residues. This shortcoming should not deter the modeler from using this repository. Instead, the results should be used only as a starting point for the construction of the protein model of interest. Another section of SWISS-MODEL[42] provides the ability to construct protein models based on a user-defined target sequence and templates. Protein models constructed with SWISS-MODEL[42] allow for varying degrees of user interaction and provide

the ability to construct the models with multiple templates (up to five template structures). Multiple template modeling[43–48] is discussed later.

BLAST and PSI-BLAST

The Basic Local Alignment Search Tool[37,38] (BLAST) is a search method for analyzing the sequence of interest and for locating potentially similar protein sequences. In the chain of events discussed here, the BLAST search takes the user-supplied sequence or the sequence from a repository and searches for similar sequences and structures. Performing a BLAST search is a combination of searching for similar structures and sequences and sequence alignment. The BLAST program has evolved and been incorporated into websites and different genomic tools, thus becoming a benchmark on which other search tools are compared.

BLAST has been a staple for database searches and alignments of nucleotide and protein sequences since its introduction. Using a heuristic search methodology, a database of sequences is scanned and compared by BLAST with short segments of the target sequence. When one or more possible sequences with a short segment corresponding to the segment of interest are found, these short segments serve as the origin for the alignment process. The probability of expecting a false positive (nonrelated sequence) is calculated. This value (termed the expected value) is simply a numeric indicator of the statistical significance of the target's alignment to a given sequence found in the database being searched and will be discussed later. In this section, the uses of BLAST, as it relates to proteins, is discussed along with an overview of its methodologies.

BLAST searches are flexible in their ability to locate various types of sequences. The National Center for Biotechnology Information[49] (NCBI) has an extensive BLAST server[50] capable of searching nucleotide, protein, and translated sequences, along with sequences relating to gene expression, immunoglobulin, vector contamination, and genomes of specific species, to note a few. The protein sequence alignment provides several different databases to search with different methodologies. The NCBI-BLAST service currently maintains seven protein databases providing the ability to initiate searches ranging from being broad in scope to being highly focused. The most general database is the Non-Redundant (NR) database that contains the sequences of proteins from the PDB,[36] Swiss-Prot,[28,29,51] Protein Information Resource[52] (PIR), Protein Research Foundation[53] (PRF), and those translated from GenBank.[54] In the NR database, repeated sequences are removed to obviate multiple results from the same sequence to reduce search times. There is a supplemental nonredundant database that draws from the same sources as does the full nonredundant database. It contains only entries that have been released in the past 30 days, thus providing a quick way to search for new additions. Three lesser known sequence databases are available: the NCBI

reference sequence project, proteins registered in the patent division of GenBank,[54] and, sequences translated from environmental samples (uncultured bacteria from soil and marine samples).[55] The sequences translated from the environmental samples are not included in the previously mentioned NCBI NR database. Although the NR database contains a large body of information, it can search only the current nonupdated Swiss-Prot[28,29,51] or the PDB[36] sequence databases. The sequence database of most interest to those constructing protein models is the PDB because it provides the most likely template candidates. Just as there are different types of sequence databases, tuned versions of BLAST also exist. At the NCBI, there are several BLAST methods for locating homologous proteins. The preferred method of searching for homologous proteins is the basic BLAST search. It employs the methodologies outlined in the original BLAST paper.[37] Where searches are based on local regions of similarity, and, in cases involving similarity across the entire sequence, a global alignment can be done. The other BLAST search methods are more restrictive but can be more useful. The most sensitive of these "narrow parameter" search methods is Position-Specific Iterated (PSI)-BLAST,[38] which is adept at locating distantly related proteins. PSI-BLAST is an iterative process that starts with the results of a basic BLAST search and uses that alignment to construct a position-specific scoring matrix (PSSM) from the sequences with *E*-values (the probability that the corresponding sequence is a false positive, which is discussed in detail later) better than the user-defined inclusion tolerance. The ability to include or to exclude sequences obtained from a search is possible but requires user intervention before construction of the next profile. A profile is a numerical representation of the sequences that includes information about its physical attributes. A PSSM is a profile; it is the probability of a specific amino acid residue occupying its position in the sequences based on related sequences. The nth PSSM (also known as a profile) evaluates the alignment of the n+1th search iteration. Unique sequences (compared with the previous searches) with *E*-values better than the inclusion threshold are grouped into the PSSM sequences and a new profile (PSSM) is constructed. This iterative process continues until searches return no additional sequences based on the inclusion threshold or by user intervention. The PSSM that is created is portable between sequence databases when searching for related sequences. The Pattern-Hit Initiated (PHI)-BLAST is another narrow parameter search program that uses (1) a user-defined amino acid residue pattern and (2) the sequence containing the residue pattern of interest. The PHI-BLAST search locates sequences having the pattern of interest, (i.e., they are similar in the region of the pattern) but that might not have any homology with the original sequence. BLAST can be used to search for short, nearly exact sequence matches for small peptides (typically 15 residues or less) when seeking a specific residue pattern. To search for a conserved domain, one can use the reverse position-specific (RPS)-BLAST method that is based on PSI-BLAST. It does not require iterative searches; instead, RPS-BLAST searches the conserved domain

database (CDD) of predetermined PSSMs to find conserved domains rather than generating the PSSM as a way of refining the search results. The NCBI's CDD is a collection of PSSMs based on sequence alignments contained in the following databases: Simple Modular Architecture Research Tool[56,57] (SMART), Pfam,[58] Clusters of Orthologous Groups of proteins[59,60] (COG), and Library of Ancient Domains (LOAD). The RPS-BLAST search is also a quick way to discover the biological function of the protein being studied. The final protein BLAST implementation of interest is the Conserved Domain Architecture Retrieval Tool (CDART). Working in conjunction with RPS-BLAST, CDART aids in the retrieval of proteins containing one or more domains in common with the sequence of interest. The RPS-BLAST and the CDART search methods are more sensitive than is the standard BLAST search. The many different search methods and parameters can be at times overwhelming. However, the wide range of search strategies now available provides a better understanding of the protein from the perspective of general homology to specific regions or domains. Several Web servers are available to perform BLAST searches. Our choice to focus on the BLAST servers at the NCBI for this chapter is due to the large number of features available to the modeler to assist in protein sequence similarity searches.

The mechanics of a BLAST search involves three steps: (1) assemble a list of probable sequences, (2) search those probable sequences for similarities, and (3) attempt to extend the regions of similarity. The list of probable sequences is, initially, all sequences in the selected database. The initial set of protein structures is thus constructed by scanning the entire database for sequences that are similar to the protein sequence of interest. The initial scan uses a short segment of amino acid residues from the target (typically three residues) along with a similarity matrix to exclude sequences that are not likely to be similar based on a preset threshold. The subset of sequences is examined for specific short segments. When a segment is found, an attempt is made to extend it in one direction until the maximal segment pair (MSP) score is less than the score for a shorter segment. Calculated heuristically, the MSP is a measure of local similarity between any two sequences. The MSP is defined as the best scoring pair of equal length residue segments between two sequences (which can be of any length). The segment pair is considered optimally aligned if increasing or decreasing the length of the segment cannot improve the MSP score. The sequences are then ranked based on similarity.

Improvements to BLAST have focused on three key areas, including a more stringent criterion for the extension of similarities, use of gapped alignments, and implementation of the PSI-BLAST method discussed.[38] One can achieve an increase in speed and number of possible sequences with BLAST by implementing the "two-hit" search method.[38] Here, a short segment from the target sequence is initially used to scan the database. If a match is found, the sequence is scanned again for a second short segment within a specific distance of the first segment. Provided that the first and second segments

do not overlap, and they are within the required distance constraint, the sequence becomes a candidate for extension. Initially, single hits are located. An extension is only allowed when an associated second hit is found. Adding a gap to an alignment can be done when the normalized score of a high-scoring segment pair, from an extension of the two-hit method, is at a preset minimum value. The newly gapped alignment is retained if its expected value is significant.

The expected value (typically reported as *E*-value or e-value) assigned to each returned protein sequence is the probability of that protein being a false positive (a protein sequence that is considered to be related, and yet is not). The *E*-value is a gradient-type, relative-value scale based on the size of the database being searched. *E*-values assigned to sequences between zero and one have a low probability of being found by chance and are considered to be significant with respect to the query sequence. Contrarily, sequences with *E*-values greater than one (there is no maximum *E*-value) might be closely related to the target protein, but they are more likely to be chance occurrences. The *E*-value is a quick way to determine the probability of a good match between the target sequence and those in the database being searched. The threshold value mentioned earlier specifies the maximum number of significant sequences that will be returned. Increasing the threshold value will increase the number of divergent sequences returned.

BLAST is an invaluable tool for sequence and template searches and for basic sequence alignments. The BLAST search method is ubiquitous, and a "Quick BlastP Search"[38] button is available at the top of Swiss-Prot sequence entries. This button will initiate a standard protein–protein BLAST search (commonly denoted as blastp) for related protein sequences. The diverse set of tools available range from very basic sequence searches and domain classification to searching for very specific residue segments. An adaptation of the original BLAST methodology, the Washington University (WU)-BLAST,[61] developed and maintained by Warren Gish, is purported to improve sensitivity and speed.

Protein Data Bank (PDB)

The PDB[36] is the repository for solved 3-D structures of proteins, peptides, viruses, protein-nucleic acid complexes, nucleic acids, and carbohydrates. Most structures deposited in the PDB are based on X-ray crystallography (approximately 80%) with the remaining structures solved using nuclear magnetic resonance (NMR) techniques. The PDB no longer stores theoretical protein models in the main archive, but it does accept them for storage in a separate theoretical model section where they are neither annotated nor validated. The PDB staff members annotate and validate the submitted 3-D structures to ensure the information provided to the public is correct. The structures archived at the PDB are provided in two formats, PDB and

macromolecular Crystallographic Information File[62,63] (mmCIF), both of which contain the same information. Any structure obtained from the PDB contains information about the source of the protein structure, chains (sequences) that compose the structure, cofactors and prosthetic groups with chemical formulas, component names, qualitative descriptions of the protein's and crystal's characteristics, article reference(s), and the Cartesian coordinates.[36] With the exception of the resolution of the structure (if determined using X-ray crystallography), this basic information is typically the most important when considering the selection of a template structure. Other information regarding the conditions and the methods for solving the protein structure is also included. This information includes the methods and conditions that crystallize the protein (temperature, pH, solvents, and salts), the occupancy and temperature factor for each atom, unit cell dimensions and space group assignment, diffraction data collection methods, and the refinement data (specifically the method of refinement, *R*-factor, and resolution limits). NMR-based protein structures are occasionally used for a template, and it is important to realize that these structures are *dynamic* in contrast to X-ray structures, which are derived from a rigid lattice. The NMR structures contained in the PDB have information about the number of models in the ensemble and if one of them should be considered to be a representative structure or an energy minimized average structure. These NMR-derived structures also include information similar to the crystallographically derived structures. Specifically, information regarding the experimental conditions and parameters (methods, magnetic field strength, probe head, and sample tube) are given along with information about the NMR experiments conducted and the restraints used to solve the protein structure. All of this information is important to the modeler to determine whether the structure is suitable as a template.

Several methods exist for searching the PDB depending on what is known about the system of interest. These search methods include PDB ID, QuickSearch, SearchLite, SearchFields, Search Status, and Iterative Search, each of which will be briefly discussed. The searches are case insensitive (HUMAN LYSOZYME is the same as human lysozyme or Human Lysozyme), and results can contain one or multiple structures. As noted, the PDB is an archive, and the amount of information about one particular structure may not be as in-depth as for other similar structures based on methods and information available at time of deposit. The PDB is currently updating the information of older submissions to ensure a uniform dataset between all entries. All structures deposited to the PDB are catalogued by a four-character identification name designated at the time of deposit. Using this PDB ID code is the simplest way to find the protein structure of interest. If the unique identifier is not known, another search method can be employed. The QuickSearch method allows for a text search of the structure files and webpages containing information related to the search query. The SearchLite method searches the PDB for the word (or phrase) as entered (QuickSearch

also uses this method). Entering "human lysozyme," for example, would initiate a search for the phrase "human lysozyme." Structures (or webpages) with human and lysozyme occurring separately will not be returned. However, a Boolean search using the key words "and," "not," and "or" can be invoked to locate specific structures or a group of structures. To perform a search for structures with both human and lysozyme in the entry, for instance, the query would need to be "human *and* lysozyme." The SearchLite method can search for a portion of a word, allows the removal of structures with the same sequence, and searches for the structure under consideration. The SearchFields option is the most robust of the four search methods available, which allows for the addition of search fields, thus permitting the user to fine-tune searches. The basic fields are PDB ID, text searches, chain type (protein, enzyme, glycoprotein, carbohydrate, DNA, RNA, and DNA/RNA hybrid), and so on. Fields available to refine the initial search include general information (author, EC number, ligands and prosthetic groups, and the source of the protein), sequence and secondary structure features (FASTA, short sequence patterns, and secondary structure content), and experimental crystallographic information (resolution, space group, unit cell dimensions, and refinement parameters). The Status Search is unique because of the search fields available. In addition to the common fields like PDB ID, author, and title or name of the structure, additional search fields related to the status of the to-be-released structures, the sequence availability, and the release and deposit dates are provided. The search results from QuickSearch, SearchLite, and SearchFields can be further refined by iterative searches or by removing structures with similar sequences.

Sequence Alignment and Modeling System With Hidden Markov Models

The Sequence Alignment and Modeling[64–66] (SAM)-T02 system is a suite of applications that provides, among other things, a sequence alignment of the target to all possible templates, predictions of the target's secondary structure, a list of the most probable templates and their alignment to the target, and a 3-D model. It is important to note that SAM-T02 is a comparative modeling server and the underlying program that constructs the protein models is UNDERTAKER.[65] Here we focus on SAM's ability to locate similar sequences and possible templates. SAM searches can detect evolutionarily distant proteins (providing results similar to those of a BLAST search) with the addition of a predicted secondary structure (to improve alignments) and a listing of possible fragments that can be used for the construction of a target model using either Segment Match Modeling[5,67] or Multiple Template Modeling[43–48] (both methods are discussed later). There are six steps involved when using SAM-T02 to build a protein model: (1) Find sequences similar to the target via iterative searches, (2) predict the secondary structure with an artificial neural network, (3) use two-track hidden Markov models[1,68–71]

(HMM, discussed later) to find probable templates for threading (fold-recognition, discussed later), (4) align the target with the templates, (5) construct a fragment library for the target with FRAGFINDER, and (6) build a 3-D model of the target through fragment packing and threading (fold-recognition alignments) with UNDERTAKER.[65] SAM-T02 provides the user with the ability to improve the alignment of a sequence and the selection of a template, both of which are important in constructing a quality protein model.

Homologous sequences can be located by using four rounds of WU-BLAST[61] with increasing threshold values (0.01, 1.0, 10, and 400).[64] An artificial neural network is used by SAM-T02 to predict the secondary structure. These results are presented in one of five secondary structure notations (alphabets). The neural networks that are used can predict the secondary structure occurring at each residue's position in the sequence based on the target's HMM.[64] A probability distribution for the likelihood of each type of secondary structure for each residue is given. The prediction of secondary structure is presented in the STRIDE[73,74] notation, two variations of the Definition of Secondary Structure of Proteins[72] (DSSP), the standard EBGHSTL and the expanded STR, which incorporates six different β-sheet classifications (Table 1),[66] and an 11-state notation representing the torsion angles of four consecutive Cα atoms[66] (the results of the Cα are not currently provided on the SAM-T02 Web server, but they are reported when using the downloadable version). The DSSP method is a standardized method of defining secondary structure of a solved protein structure based on hydrogen bonding and geometrical features; the PDB uses DSSP for secondary structure determination of deposited protein structures. It should be noted that these methods do NOT predict secondary structure. STRIDE assigns secondary structure to a protein

Table 1 Secondary Structure Notation

		Secondary Structure Notation		
	Secondary Structure	DSSP[72]	STRIDE[73,74]	STR[66]
α Helices	α helix	H	H	H
	3/10 helix	G	G	G
	π helix	I	I	H
β Sheets	Extended (↑)	E	E	E
	↑↑↑	E	E	P
	↓↑↓	E	E	A
	↓↑↑	E	E	M
	↑↑	E	E	Q
	↑↓	E	E	Z
Undefined	Isolated β-bridge	B	B or b	B
	H-bonded turn	T	T	T
	Bend	S	–	S
	Coil	–	C	–

structure based on hydrogen bonding energy and backbone torsional angles. The STR method expands the extended β-sheet notation (E) used by DSSP and STRIDE into six types of β-sheets depending on the neighbors of the β-strand of interest. For STR, the E notation indicates a β-strand that has no corresponding neighbors. In cases in which a β-strand has two neighbors, the direction (the carboxyl group of a residue is considered the leading end of a peptide) of the neighbors is incorporated into the type of β-sheet secondary structure. The parallel designation P is when both neighboring residues travel in the same direction as the residue of interest (↑↑↑), A signifies two antiparallel neighboring residues (↓↑↓), and M denotes one antiparallel and one parallel neighboring residue (↓↑↑). When β-strands are the edge of the defined secondary structure, they only have one neighbor that can be parallel (↑↑) Q or antiparallel (↑↓), Z. Additionally, for the Critical Assessment of Techniques for Protein Structure Prediction[75] (CASP) assessment, the secondary structure is returned in the reduced helix, sheet, and coil notation. Templates are located by SAM-T02 by using the single two-track target HMM together with the STR secondary structure predictions. Twenty-five different alignments of the target to each template are devised with various alignment preferences and different target- and template-based HMMs. Next, the fragment library is constructed with the two-track HMM based on the STR secondary structure prediction. The fragment search seeks the best six gapless fragments (with a length of nine residues) for each residue position in the target sequence. The reliability of the fragment library depends on the validity of the HMMs used in the secondary structure prediction; probable but incorrect predictions will initiate poor fragment selection. The fragment library, the alignments, and a generic fragment library are all used to construct a 3-D representation of the target protein using multiple-template and threading methodologies. The generic fragment library is composed of one-, two-, three-, and four-residue fragments collected from a training set of 448 monomeric protein chains.[65] The alignments and the corresponding fragments provide an initial model that is subjected to many iterations of genetic algorithm-based minimization and energy function evaluation. Unlike typical multiple-template methods (discussed later), noncontiguous backbones are welcomed and allow the user to gather multiple-segment information from the threading and fold-recognition alignments. These features of SAM-T02 allow one to sample from many different alignments and combine parts of various alignments, which is advantageous when working with distant target-template relationships. When the target and template(s) are similar, it is beneficial to select the best alignment, thus focusing the search. In contrast, for distant relationships, the mixing of different alignments can be useful for locating suitable segments. The SAM-T02 server is constantly evolving in an attempt to provide the most comprehensive and best protein structure predictions. Some of the SAM-T02 methods discussed here are not available on the developer's website and require the download and installation of the SAM-T02 application on a local computer.

The SAM-T02 method relies heavily on the HMM[1,68–71] method, whereas the transmembrane (TM)HMM secondary prediction method[71,76,77] (parameterized for G protein-coupled receptor and transmembrane proteins) uses HMMs to predict only the secondary structure. A HMM is a statistical modeling method used in the alignment of multiple amino acid sequences and other bio-informatics applications in addition to protein secondary structure predictions. It is similar to evolutionary programs[78] in its ability to be plied into the tool needed for the job; yet it closely resembles an artificial neural network[79] (ANN) that is typically used for constructing quantitative structure-activity relationship (QSAR) models. HMMs can be trained to predict the secondary structure for specific families of proteins. The HMM's similarity to ANNs comes from its need to be trained on a training set of data.

HMMs are considered the most sensitive method of detecting sequence similarities and are constructed from a series of states (known observations). An HMM is a profile, but it is more complex because of its use of advanced topology. The HMM works by progressing through a series of states, which produces a result when a given state is reached, or by moving from state to state,[80] similar to a finite state machine. When using an HMM for the prediction of protein secondary structure, the probable secondary structure type is output as the HMM moves from state to state. The secondary structure is selected based on an ''emission probability'' table (similar to a profile) with the added probability of moving from state to state. The HMM method differs from profile-based methods of predicting secondary structure because of how it assigns gap penalties. The gap (or insertion) penalty is the same in a profile-based method, whereas the penalty can be varied depending on whether the region is highly conserved or varied in an HMM method.[81]

HMMs have also been used search a database of protein sequences and for the alignment of multiple sequences. The SAM-T02[65] system is a secondary structure prediction server using HMMs to predict the secondary structure of protein sequences. The predicted secondary structure of bovine α-lactalbumin as predicted by the SAM-T02 server is presented in Figure 3 in the STR, DSSP-EBGHSTL, STRIDE-EBGHTL, and DSSP-EHL2 secondary structure notations, respectively. The predicted secondary structure can be used in conjunction with or aid in the alignment of the target to the template amino acid residue sequence.

Threading

The selection of a template typically follows BLAST-type searches and sequence alignments. The template selection is based on the similarity of sequences but neglects the possibility that templates with a similar structure may have differing protein functions. Threading provides a way to account for the possibility that functionally different proteins share similar structures. Instead of matching the target sequence to all possible sequences (with or

without a known 3-D structure), threading (1) creates pseudo-protein models based on solved protein structures, (2) calculates an energy value for that pseudo-model using an empirical energy function, and (3) ranks the alignments based on that energy. The lowest energy value for the pseudo-model is considered the most probable template. The threading concept is depicted in Figure 4. The pseudo-protein model is not a complete protein model. In a pseudo-protein model, the known protein structures are used as a scaffold on which the target sequence is placed; the residues, however, are simply points in space, and the model resembles a folded strand of beads. The energy function is a knowledge-based, pairwise potential that is parameterized from known protein structures. The energy of each pseudo-model is the sum of

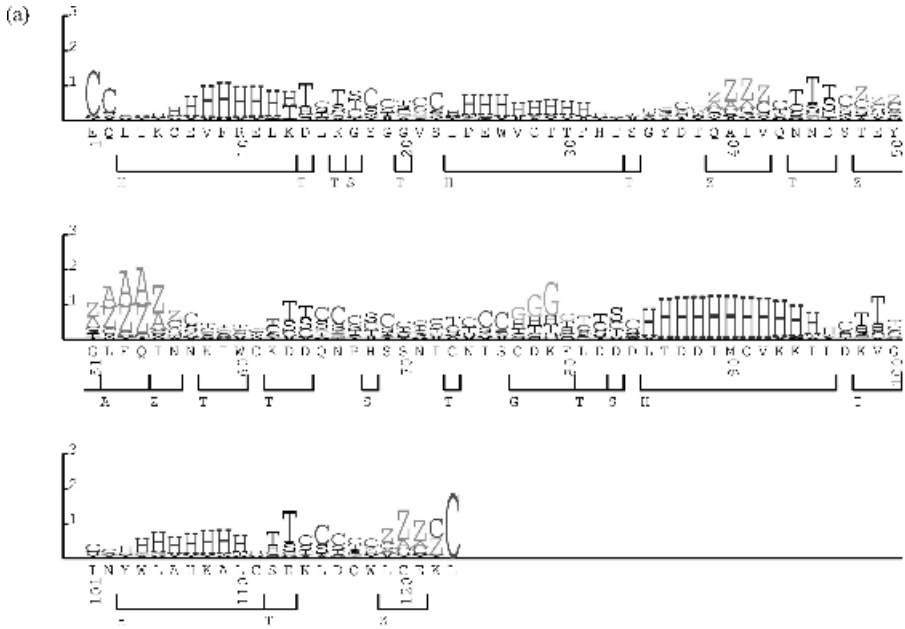


Figure 3 SAM-T02 results for the secondary structure prediction of bovine α-lactalbumin. The predicted secondary structure of proteins by SAM-T02 is reported using four different secondary structure evaluation methods. The relative size of the letters corresponds to the reliability of the predicted secondary structure. For bovine α-lactalbumin, it leads to residues with only one type of predicted secondary structure (specifically, large C's or H's) or residues with several secondary structure types (specifically, T's, S's, C's, and X's). The brackets under the residues signify the type of expected secondary structure given the predictions. The predicted secondary structure is reported in the STR (a), DSSP-EBGHSTL (b), STRIDE-EBGHTL (c), and DSSP-EHL2 (d) formats. Analysis of the results indicates the predicted secondary structure is similar to that of chicken lysozyme or human α-lactalbumin.

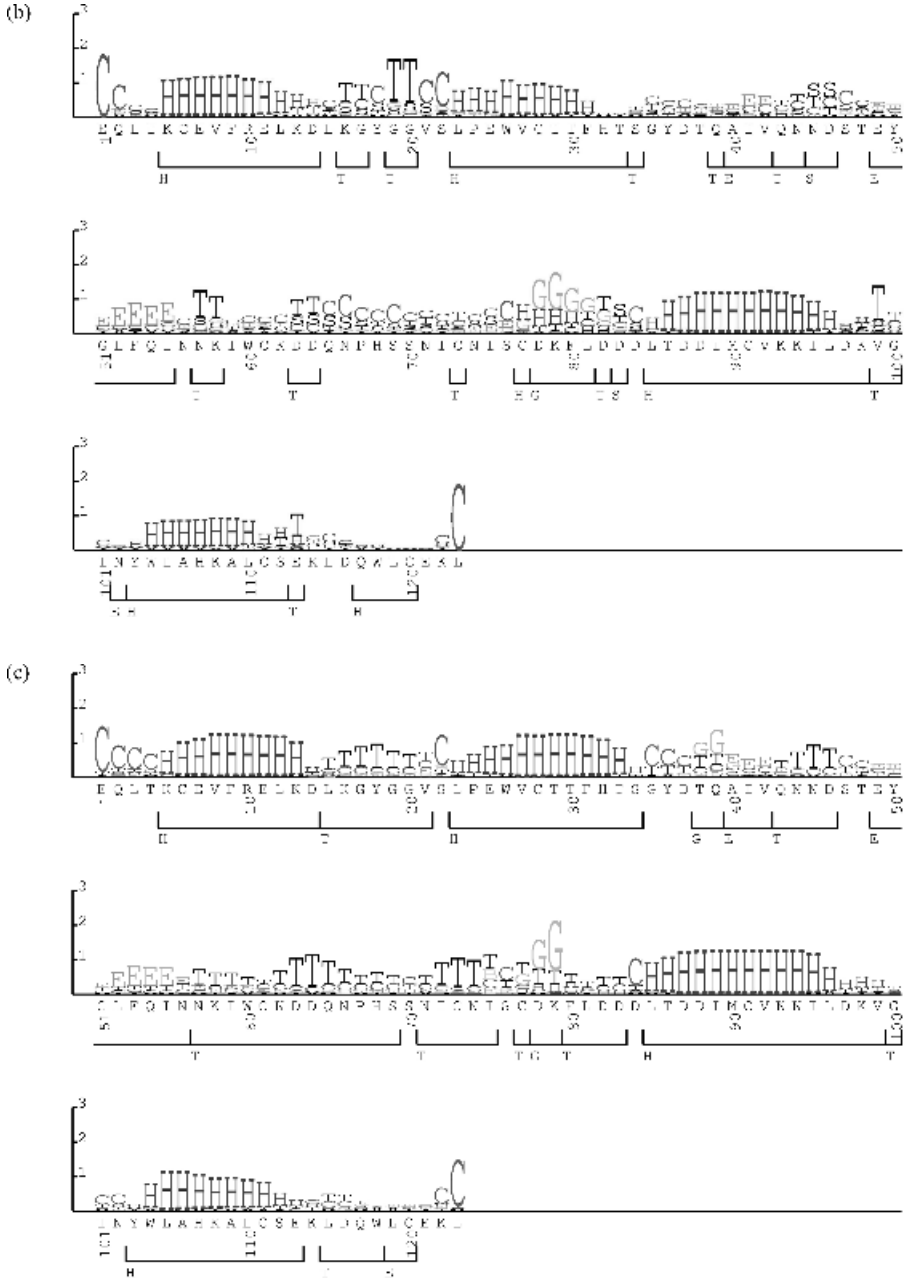


Figure 3 (*Continued*)

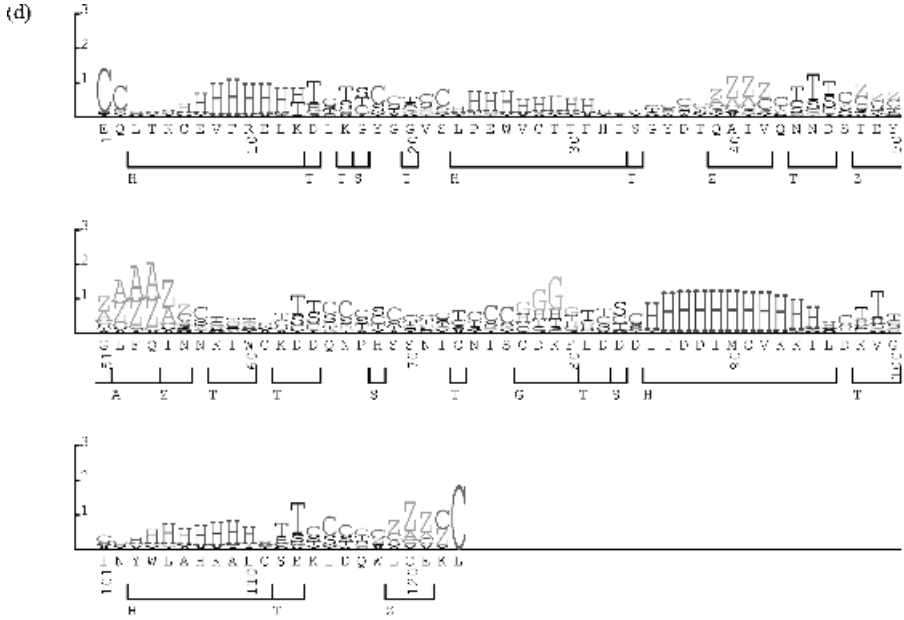


Figure 3 (*Continued*)

residue interactions. The main differences between most existing threading methodologies lie in the construction of the empirical energy function and how the knowledge base is devised.

Using threading methods, many possible configurations can be examined. Then, a *Z*-score function quickly evaluates each possible configuration. The *Z*-score is a normalized value that indicates the standard deviation for a specific data point (pseudo-model) compared with the mean of the dataset. The drawback of using a simple scoring function like the *Z*-score is its low resolution (i.e., that the predicted structure will not be able to match the known X-ray structure). It in turn leads to indecision when trying to determine which of the top-ranked pseudo-structures is the best template to use. The underlying principle of a threading scoring function is that the native sequence threaded onto its own scaffold will result in a lower energy than if it is threaded onto any other scaffold. It is thus beneficial to the modeler to have a database containing several known 3-D structures from each protein family to provide a variation of pseudo-structures. This way, by threading the target sequence onto all possible templates, the lowest energy pairing should be the most likely template to select. Supplementing threading methods with information about secondary structure, solvent exposure, or residue burial can improve the final alignment, especially when gaps are considered.

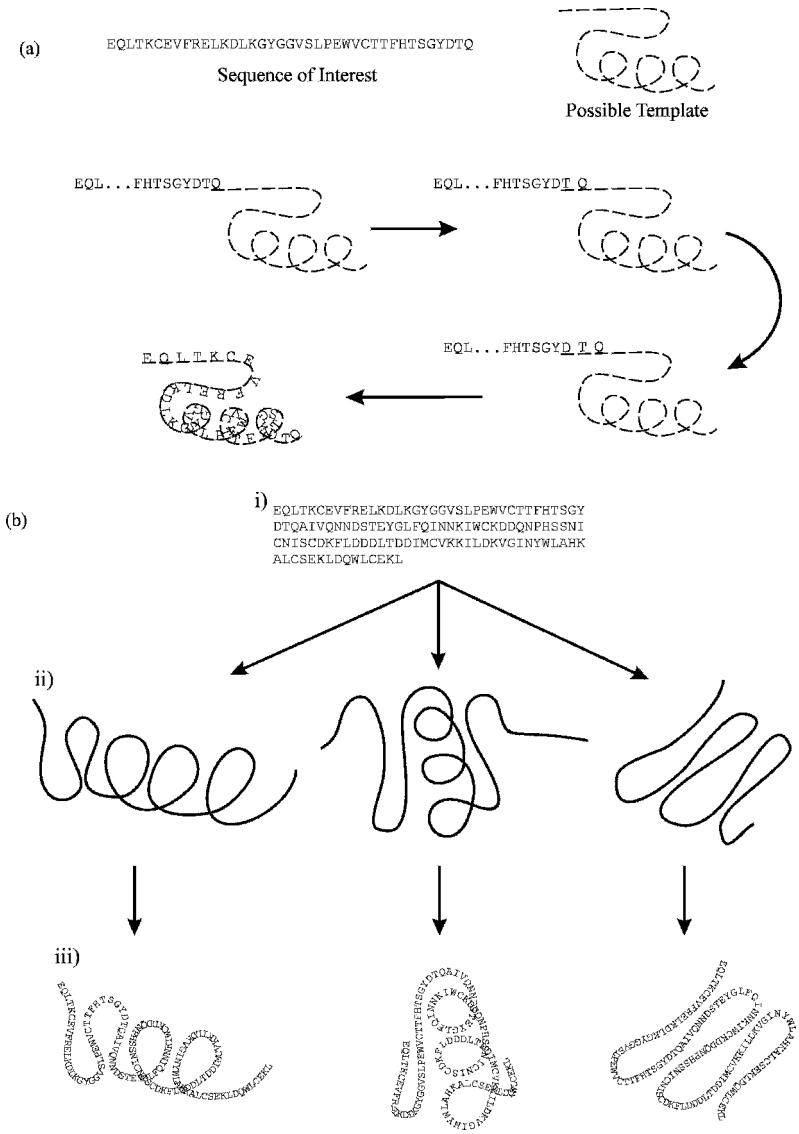


Figure 4 Template searching via threading. The sequence of interest is "threaded" onto a possible template one residue at a time in (a). The incremental addition of the residues to the template creates many possible models (pseudo-protein models) for that target-sequence combination. Each model is evaluated, and the best pseudo-protein models for the grouping (alignment and energetic information) are stored. Once the pseudo-protein models are created for all possible templates, the proposed alignment and templates are ranked and selected based on a combination of *Z*-scores, energetics, number of aligned residues, and percent identity and/or similarity. (b) illustrates the stepwise process of threading the target sequence i) to several possible templates, ii) thus creating several pseudo-protein models iii) that will be evaluated and ranked.

The power of threading exploits the fact that proteins with different functions can possess similar structures even though they may have little to no sequence identity or similarity. THREADER[82–84] and Learning, Observing, and Outputting Protein Patterns[85–87] (LOOPP) rely on similar strategies, yet they use different energy and scoring functions to generate probable alignments with feasible templates. To illustrate the similarities and differences of these methods, we now compare and contrast them.

THREADER

Like other threading methods, THREADER uses solved protein structures as a scaffold on which to place the target protein's sequence. Secondary structure information (from secondary structure predictions, NMR, circular-dichroism, or other experimental methods) about the target sequence is used to force the alignment between the predicted secondary structure of the target with that of the actual secondary structure of the known proteins. Because energetic functions cannot differentiate between properly and incorrectly folded protein structures, THREADER uses a set of knowledge-based potentials to indicate misfolded proteins. The potentials used by THREADER are derived from statistical data compiled from known protein structures, and the pairwise pseudo-energy is computed using the method of Sippl.[88] In an iterative process, as each residue is placed onto the scaffold, a knowledge-based potential determines the worthiness of the pseudo-protein (target–template alignment). The interaction energy terms are divided into three groups based on overall topography of the protein. The short-range potential terms assess the probability of secondary structure on a local scale, as, for example, predicting whether a segment is an α-helix. The medium- and long-range terms (different than those described by Sippl[88]) assess the energetics of local secondary structural motifs and protein packing, respectively. Interactions exceeding 10 Å are handled by a potential function that assesses the amount of solvation associated with an individual residue (based on an approximate solvent-accessible surface area) as the target residue is incorporated into the template. The potentials used to evaluate the overall structure are also divided, based on the type of secondary structure of the template. Because the loops of a protein are often the least conserved and most solvent-exposed, they are evaluated with only the solvation potential. THREADER gives the user statistical data and the pseudo-energy of the best alignment in addition to other proposed alignments between the target and the template.

Learning, Observing and Outputting Protein Patterns (LOOPP)

LOOPP[85–87] is similar in philosophy and strategy to THREADER, but it differs in its implementation of an empirical energy function and its scoring method. The most notable aspect of LOOPP is its extensive parameterization

that is based on structures from the PDB[36] and a database of close to 5 million decoy structures[85] that were created with MONSSTER.[89] Three novel implementations of common protocols (pairwise contact model, gap penalties, and *Z*-scores) differentiate LOOPP from other threading methodologies.

The creation of a new pairwise interaction model (empirical energy function) was envisioned by the authors of LOOPP as the key to devising a truly novel threading algorithm. Two main types of empirical energy functions exist: (1) those that use pairwise residue contact energies for residues within a specified distance of one another and (2) those based on the environment of an amino acid residue at a point in the structural lattice (this feature is also referred to as a profile). The energy of the total system is calculated the same way for both kinds of functions and is just the summation of the individual energy values. The authors of LOOPP also developed the Threading Onion Model[86] (THOM) algorithm, which incorporates both a pairwise interaction and an environment term into the empirical energy function. The first rendition of THOM (THOM1) focused on an individual residue and determined (1) the number of neighboring residues in the scaffold (based on a cutoff distance) and (2) the type of residue at the point of interest. The empirical energy value is calculated for each residue threaded onto the scaffold and then summed to give the total energy of the system. THOM2 improved the environment term of THOM1 by accounting for the contact between structural sites (pairwise interactions). THOM2 uses a two-degrees-of-separation parameter for the residue of interest. THOM2 was selected as the empirical energy function in LOOPP instead of THOM1 because of its ability to better predict the environment of an individual amino acid residue, thus bringing it closer to being a pairwise interaction model. The improvement of the environment term permits users to thread models (with gaps) that are considered to be the optimal alignment. The authors of LOOPP devised a simple method for assessing a penalty for the insertion of, and extension of, a gap. LOOPP uses default penalty values that the user can modify. A gap is parameterized like the other amino acid residues, thereby removing any bias regarding its insertion. The ''gap residue'' parameters were optimized with the values determined for the 20 native amino acid residues. The rankings of the threaded structures in LOOPP are based on a double *Z*-score (global and local *Z*-scores) plus the value of the empirical energy function. The global and local alignments are used to calculate the double *Z*-score value. The global alignment is closely related to the overall length of the target sequence as related to the template scaffold and is affected by differences between the target and the template. The local *Z*-score measures the fitness of the alignment between the template sequence compared with the defined secondary structure of the scaffold. The combination of a dual-layer empirical energy function and *Z*-score analysis provides LOOPP with the ability to select templates and provide informative alignments accurately. The use of a double *Z*-score filter aids in

retaining more correctly predicted alignments and templates than when using a single *Z*-score, and it does so while reducing the number of false-positive templates.

The databases (repositories, archives, sequence retrieval systems) discussed in this section are a small sampling of those databases currently available. These resources were discussed only in a cursory manner, highlighting their content, the search methods available, and their importance in the search for sequences and structures. The primary function of the PDB is to validate, annotate, and archive the solved 3-D structure of macromolecules for the scientific community. The search features available are not as broad as those of ExPASy (neither is the amount of data housed), but the search tools provided in the PDB are robust for the information contained in that database. It is easy to see why specialized proteomic databases are needed; the myriad of information is growing rapidly, and the ability of a few large database centers to manage all that data is becoming unrealistic.[23] The use of threading to aid in the selection of possible templates and for target-template alignments is a powerful methodology. The results obtained from THREADER and LOOPP (or other threading methods) are excellent starting points for locating a group of likely templates or to provide additional information for an already established alignment. Threading brings to light relationships and information that might have otherwise been overlooked because of evolutionary distance and differences in the function of the systems being used as templates.

Example: Finding Related Sequences and 3-D Structures

The example presented here involves the construction of a protein model for bovine α-lactalbumin, a revisit of the original comparative protein model study of Browne et al.[3] described in the Introduction of this chapter. It is not an attempt to reproduce or refute any of their results but instead is a simple example for novices to the field of comparative modeling. The first step is to find related sequences and structures. It should be noted that there are several solved 3-D structures of BCLA. It might cause some problems when searching for related templates; yet the structure of lysozyme (used in the Browne et al. study) is also available as a possible template. An ExPASy search for "bovine lactalbumin" on www.ExPASy.org provides a wealth of information including links to the solved 3-D structures and the primary sequence (ExPASy access code P00711). α-lactalbumin is 142 residues in length with the signal sequence being the first 19 residues. A PDB "bovine lactalbumin" search on www.rcsb.org returns no hits, but when shortened to "lactalbumin" provides 22 structures from various sources including human, bovine, and goat. Resubmission of the PDB search using "bovine *and* lactalbumin" returns 16 structures for BCLA. The 16 structures returned by the PDB includes the three structures noted in the ExPASy report with 13 additional structures that are either not from bovine (*Bos taurus*) or were combinations of bovine and

another species α-lactalbumin. The complete bovine α-lactalbumin sequence (142 residues) was submitted for a BLAST search on the NCBI website. The initial BLAST search of the nonredundant database resulted in many hits, whereas a BLAST search of the PDB database gave 44 α-lactalbumin structures. Additionally, a "Quick BlastP Search"[38] was initiated from the ExPASy entry, and the 100 most likely sequences were returned.

Aside from selecting the template based solely on sequence identity, we also conducted a threading experiment to see whether other likely templates exist. Three THREADER[84] searches were conducted. The first search focused on the genomic sequence (BCLA with the signal prepeptide), and the second involved the domain sequence. We did this to illustrate the differences one would encounter when aligning the genomic and domain versions of a protein's sequence. The third THREADER search used an augmented library that included template information for human α-lactalbumin (1b9o[36,90]), BCLA (1hfz[36,91]), and horse hemoglobin α and β (1g0b[36,92]). The goal of this search was to demonstrate THREADER's ability to select and rank protein structures that we thought would be good templates. The results of our searches are presented in Tables 2–5. The templates were ranked based on their combined energy *Z*-score (primary sequence or the primary sequence with the predicted secondary structure as determined by PSI-PRED[93,94]). The pseudo-protein models were also ranked with the Threading Expert.

The initial THREADER results for the BCLA sequence were not promising; the largest *Z*-scores value was 3.28 for the domain sequence and 3.68 for the genomic search. The *Z*-scores in THREADER are partitioned into regions of significance that indicate the likeliness of a template being the correct template. *Z*-scores greater than 4.0 are considered to be "very significant," and scores between 4.0 and 3.5 are deemed "significant." Proposed templates that score between 2.7 and 3.5 are thought of as being "borderline significant," *Z*-scores ranging between 2.0 and 2.7 require another means of confirmation,

Table 2 THREADER Template Selections Using the Genomic α-Lactalbumin Sequence[†]

	Primary		Secondary		Threading Expert	
Rank	Template	*Z*-Score	Template	*Z*-Score	Template	Score
1	1akq	3.68	1udm	4.68	1baq	0.8164
2	1m7e	3.23	1s9u	3.74	1s9u	0.8093
3	1dbw	3.08	1tiq	3.44	3lzt	0.7926
4	2uag	3.00	1a4i	3.35	1gni	0.7891
5	1npu	2.87	1k2y	3.33	1jsc	0.7580

[†]Template searches in THREADER were performed with the primary sequence and the primary sequence with the predicted secondary structure as determined by PSI-PRED.[93,94] The results of a THREADER template search aided by the predicted secondary structure information can then be analyzed with the Threading Expert. The top five templates based on *Z*-Score (Primary and Secondary) and overall Score (Threading Expert) are presented. The original THREADER library was used.

Table 3 THREADER Template Selections Using the Domain α-Lactalbumin Sequence[†]

	Primary		Secondary		Threading Expert	
Rank	Template	*Z*-Score	Template	*Z*-Score	Template	Score
1	1m4j	3.28	1udm	4.19	3lzt	0.9099
2	1ep3	3.08	1chm	4.05	1cof	0.8573
3	1ft9	2.73	3lzt	3.11	1v6f	0.8180
4	1rlj	2.70	1k0r	3.08	1khy	0.7996
5	1a8y	2.68	1axj	3.00	1ukx	0.7718

[†]The signaling portion of BCLA was removed, and the original THREADER library was searched again. The template searches in THREADER were performed with the primary sequence and the primary sequence with the predicted secondary structure as determined by PSI-PRED.[93,94] The results of a THREADER template search aided by the predicted secondary structure information can then be analyzed with the Threading Expert. The top five templates based on *Z*-Score (Primary and Secondary) and overall Score (Threading Expert) are presented. The original THREADER library was used.

those with a score less than 2.0 are considered to be improbable templates, and *Z*-scores of −9.99 are an indication that a significant portion of the target sequence has not been aligned to a template. The Threading Expert is distributed with THREADER. It is used to analyze the output of a secondary structure-aided template search, and it provides a single numerical value that corresponds to the probability of the pseudo-model being the correct template. The core of Threading Expert is a neural network that is trained on correctly threaded matches (versus incorrectly threaded matches).[95] The results in Table 2 indicated that the most likely template is the apoflavodoxin-riboflavin complex from *Desulfovibrio vulgaris* (PDB ID: 1akq[36,96]) with a *Z*-score of 3.68 and having 128 aligned residues. This is troubling as it was expected that the top 20 possible templates for the genomic search would include one of the solved 3-D structures of BLCA or one of the proposed templates from the study of Browne et al. For the genomic search, THREADER did not find

Table 4 THREADER Template Selections Seeded With Probable Templates[†]

	Primary		Secondary		Threading Expert	
Rank	Template	*Z*-Score	Template	*Z*-Score	Template	Score
1	1m4j	3.28	1udm	4.19	1hfz	0.9125
2	1ep3	3.08	1chm	4.04	3lzt	0.9099
3	1ft9	2.72	3lzt	3.10	1b9o	0.9012
4	1rlj	2.70	1k0r	3.07	1cof	0.8573
5	1a8y	2.67	1axj	3.00	1v6f	0.1810

[†]The addition of several templates that are known matches for BCLA were added to the library of possible templates. The additional protein structures are HCLA (1b9o[36,90]), BCLA (1hfz[36,91]), and HBA and HBB (1g0b[36,92]). The template library was again searched against the domain sequence of BCLA.

Table 5 Proposed Bovine α-Lactalbumin Templates[†]

	Primary			Secondary			Threading Expert	
Template	Rank	*Z*-Score	Aligned	Rank	*Z*-Score	Aligned	Rank	Score
Genomic BLCA Sequence								
3lzt	190	1.62	121	12	2.80	121	3	0.7926
1a6m	267	1.44	119	1174	0.39	129	133	0.4881
Domain BLCA Sequence								
3lzt	62	2.07	121	3	3.11	121	1	0.9099
1a6m	1396	0.31	106	1776	−0.06	121	181	0.4382
Domain BLCA Sequence with Modified THREADER Fold Library								
1hfz	21	2.37	121	9	2.83	121	1	0.9125
1b9o	27	2.31	123	21	2.48	121	3	0.8573
3lzt	63	2.07	121	3	3.10	121	2	0.9012
1a6m	1416	0.30	106	1778	−0.06	121	183	0.4382
1g0bA	2422	−0.47	108	3111	−1.70	108	1972	0.1307
1g0bB	2477	−0.52	108	2276	−0.43	99	1951	0.1330

[†]The THREADER and Threading Expert results for the proposed templates for BLCA are presented for the various sequences and fold libraries used. Threading searches using only the primary sequence initially did not provide good results (original folds library); yet the inclusion of known similar folds improved slightly the overall results.

any of the solved 3-D bovine α-lactalbumin structures, and the template structures corresponding to chicken lysozyme (CLYC, PDB ID: 3lzt[36,97]) and sperm whale myoglobin (SWM, PDB ID: 1a6m[36,92]) in Table 2 were initially predicted to be poor templates. Using the primary genomic sequence of BLCA (includes signaling portion), CLYC (3lzt in Table 5) was ranked 190 with a *Z*-score of 1.62 and 121 aligned residues, whereas SWM (1a6m in Table 5) was ranked 267 with a *Z*-score of 1.44 and 119 aligned residues. Adding the predicted secondary structure, which was determined by PSI-PRED,[93,94] improved CLYC's ranking and *Z*-score to 12 and 2.80, respectively, with 121 aligned residues while reducing SWM's ranking and *Z*-score to 1174 and 0.39, respectively, with 129 aligned residues (see Table 5). Analysis of the secondary structure aided search with Threading Expert ranked CLYC in Table 5 as the third most likely template with a probability of 0.7926. Searching for probable templates with the primary-domain sequence of BLCA yielded slightly better results for CLYC (rank: 62, *Z*-score: 2.07, number of aligned residues: 121) and SWM (rank: 1396, *Z*-score: 0.31, number of aligned residues: 106). The probability of CLYC (rank: 3, *Z*-score: 3.11, number of aligned residues: 121) being a viable template improved, whereas SWM's values (rank: 1776, *Z*-score: −0.06, number of aligned residues: 121) declined again with the inclusion of the predicted secondary structure. The CLYC template was ranked first in Table 5 by the Threading Expert with a probability of 0.9099. These results were not considered to be conclusive because, on examination of the database, it was discovered that it did not

contain any α-lactalbumin structures. It prompted us to add human α-lactalbumin (HLCA, PDB ID: 1b9o[36,90]), bovine α-lactalbumin (BLCA, PDB ID: 1hfz[36,91]), and horse hemoglobin α and β (HBA, PDB ID: 1g0bA[36,92] and HBB, PDB ID: 1g0bB[36,92]) to THREADER's fold library. The modified template library was again searched with the domain sequence of BCLA yielding significantly better results (see Tables 4 and 5). Although the top five proposed templates remained the same when using the modified and the original fold library for domain-based sequences (primary and predicted secondary structure), the modified fold library had a profound effect on the results. The templates of CLYC, BLCA, and HLCA were ranked 3rd, 9th, and 21st overall with the inclusion of the secondary structure (see Table 5), and the Threading Expert ranked them 2nd, 1st, and 3rd respectively (see Table 5). These results indicate that CLYC and HLCA are viable templates for BLCA and that THREADER and the Threading Expert could select BLCA as the correct fold.

STEP 2: SEQUENCE ALIGNMENT

The alignment of amino acid residues is the most critical step in a comparative modeling study.[6] The ability to determine the best alignment of distantly similar sequences depends on several factors, including the number of sequences bridging the evolutionary distance between the target and template,[98] the similarity matrices, and the method used to align the sequences. Without a proper alignment, the models constructed will have little use and are potentially more harmful than not having a valid model at all. The use of multiple sequences in an alignment reduces the probability of chance occurrences in similarity, while increasing the chance of a correct alignment.

The goal in this part of the comparative modeling study is to align the amino acid residues of the target protein with those of related proteins (not necessarily possible templates) and proposed template(s) proteins by aligning conserved and physicochemically similar residues throughout the set of proteins. The alignment of the collected protein sequences is carried out without indicating which sequence(s) constitute the proposed template(s). Thus, the sequences are aligned based on evolutionary history and not to the proposed template. Some programs provide the option to delay the alignment (or realignment) of user-specified sequences, which thus allow one to align the target or template sequence to the already aligned sequences. The ability to delay the alignment of specific protein sequences is beneficial when working with proteins that are not closely related, thus reducing the probability of inappropriate insertions (gaps). In addition to the mechanism used to construct the alignment, i.e., the program and the methodology that are used, the parameters (evolutionary matrixes) that are used to align the sequences are also of importance and are discussed here. The programs and the methodologies

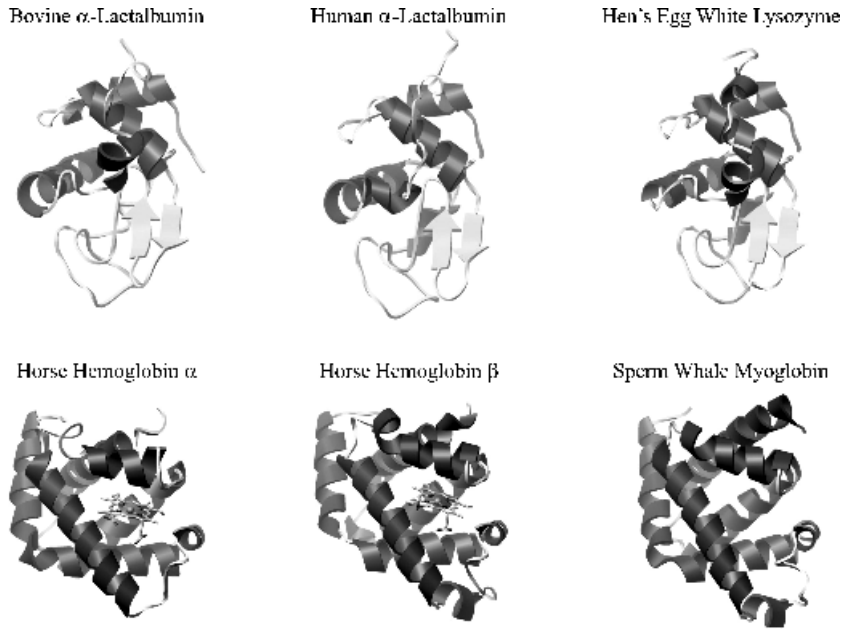


Figure 5 Structures of closely and distantly related proteins. Protein images created with UCSF Chimera.[99]

discussed here are only a sampling of the most common methods that are incorporated into websites.

In Figure 5, we see that the α-lactalbumins and the lysozyme structures possess a similar 3-D fold (overall shape) with each other as do the hemoglobins and the myoglobin. The hen's egg white lysozyme (CLYC, PDB ID: 3lzt[36,97]), horse hemoglobin α and β (HBA, PDB ID: 1g0bA[36,92] and HBB, PDB ID: 1g0bB[36,92]), and sperm whale myoglobin (SWM, PDB ID: 1a6m[36,100]) were initially considered as templates by Browne et al.[3] in their attempt to construct a protein model of BCLA. Since their landmark paper, the 3-D structure of human and bovine α-lactalbumin (HLCA, PDB ID: 1b9o[36,90] and BLCA, PDB ID: 1hzf,[36,91] respectively) have been solved (as have other α-lactalbumin, lysozyme, hemoglobin, and myglobin protein structures). Although the 3-D structure of the α-lactalbumins and lysozyme are similar (as are those of the hemoglobins and myoglobin), these proteins have different functions. The function of α-lactalbumin is to modify the substrate specificity of galactosyltransferase. It allows glucose to serve as a substrate, which thus enables lactose synthase to synthesize lactose. Parenthetically, BCLA is the cause of the human allergic reaction to cow's milk (lactose intolerance). Lysozyme's function is to initiate the breakdown of bacterial cell walls by catalyzing the hydrolysis of polysaccharides, which

thus enhances the activity of immunoagents. The hemoglobins transport oxygen from the lung to tissue. Myoglobin assists the transport of oxygen within the muscles and acts as an oxygen reservoir.

Most sequence alignment methods are automated, but some are performed manually. Both the automated and the manual alignment of multiple chains is a challenging problem because of the myriad of alignment possibilities. The number of combinations for alignment is determined by raising the number of sequences by the number of amino acid residues of the longest sequence (Equation [1]).

$$\text{Number of Combinations} = (\text{Number of Sequences})^{\text{Number of Residues}} \quad [1]$$

The alignment of two sequences with 150 residues thus has more than 1.42×10^{45} possible configurations, as an example.

The goal of automated alignment methods is to align multiple protein sequences correctly with minimal user input. Although the alignment parameters are selected by the user, they are typically the well-established similarity matrices[101–104] described here that are based on evolutionary trends. Similarity matrices are mathematical representations describing the probability of a specific amino acid residue mutating to a different residue type. The alignment programs discussed in this section differ from one another by the methods they use to align the sequences. Very similar (closely related) sequences can be quickly and aptly aligned. However, as the sequences become more distantly related based on evolution, the ability of the alignment methods deteriorates.

There are two general types of sequence alignment methodologies, global and local. Global alignments, a coarse-grain methodology, optimize the alignment between two or more sequences over their entire length. A global alignment starts at the first residue position and continues in a stepwise fashion to the last residue position. During global alignments, regions of the sequences lacking appreciable similarity are aligned to continue the search for similar regions in an attempt to align as much of the sequences as possible (quantity over quality). In contrast, local alignment, a fine-grain method, aligns regions with significant similarity first, which thus creates one or more regions of optimally aligned residues within the sequence alignment. The alignment of sequences is based on short, homologous segments (disregarding divergent regions) that decrease the number of possible alignments by focusing on the regions that are conserved over evolutionary time (quality over quantity). When searching databases of sequences, local alignments are typically used to locate regions of similarity between both closely and distantly related proteins, as in a BLAST search. The distantly related regions can be structural features (motifs or folds) or binding sites; the distantly related structural similarities are based on the configuration of several secondary structure

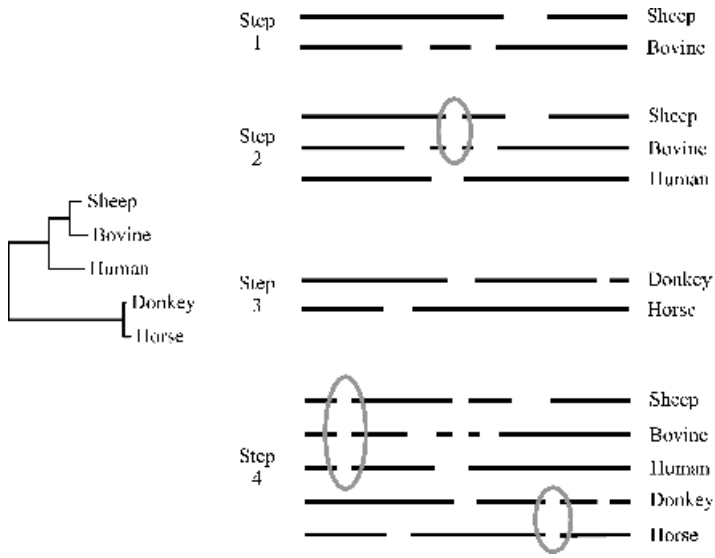


Figure 6 Progressive alignment scheme. Based on the guide tree (left-hand side of the figure), the first step is to align the sheep and bovine sequences. The second step is to align the human sequence to the previously aligned sheep and bovine sequences causing a gap to be inserted (indicated by an oval). In step 3, the donkey and horse sequences are aligned to each other. The final step is the alignment of the two groups of sequences to each other, resulting in the insertion of more gaps. The oval indicates the insertion of a gap to previously aligned sequences that was caused by the addition of new sequences. Image adapted from Leach.[1]

elements (within a region of the sequence) and the connections between them (loops). Both methods provide plausible alignments, but they focus on different aspects of the sequences.

Sequence alignment algorithms (methodologies) can be classified as being progressive (Figure 6) or fragment-based (Figure 7).[105] Progressive alignment uses iterative methods to align pairs of protein sequences or groups of sequences. The alignment is typically directed by a rooted Neighbor-Joining (NJ) tree (based on an unrooted NJ tree) using sequence weights to score the proposed alignments.[98] A fragment-based alignment method divides a protein sequence into segments (between highly conserved amino acid residues), aligns these local regions, and then aligns the connecting portions to produce an optimal alignment.

Preparing the Sequences

Before one aligns protein sequences, it is prudent to inspect the target and template sequences because one often finds the sequence of the template structure to be different from that obtained by ExPASy.[27] These differences

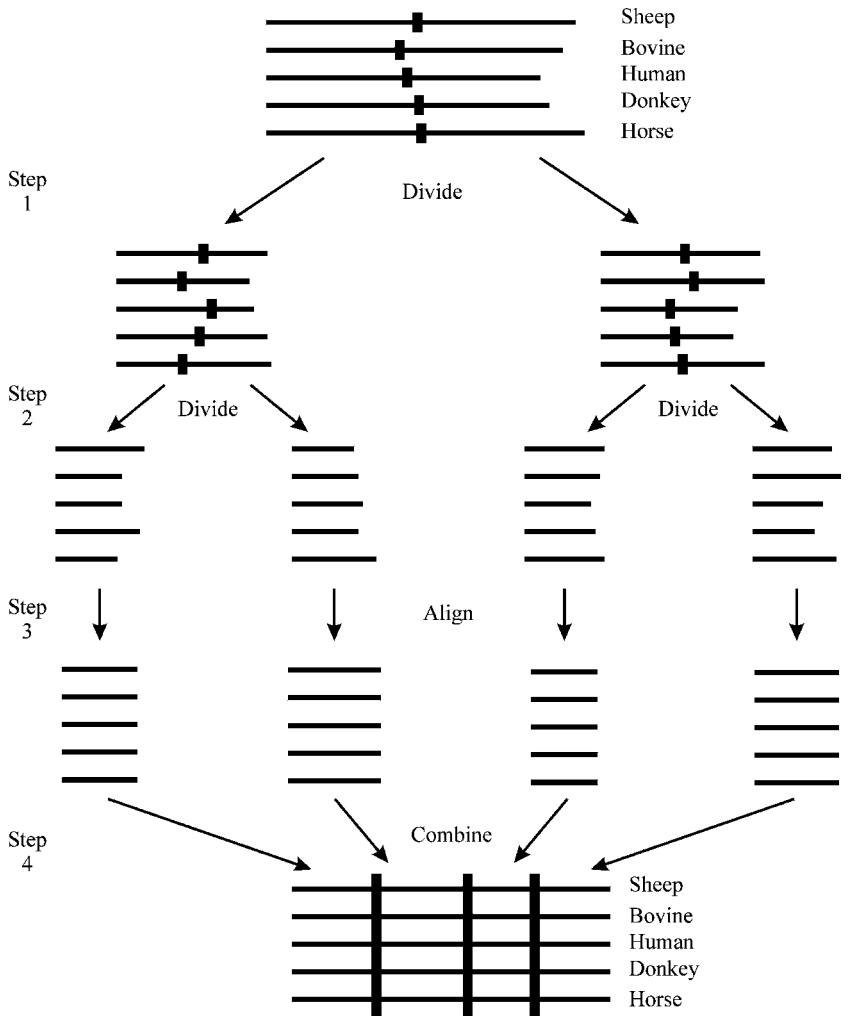


Figure 7 Fragment alignment scheme. The first step is to divide each protein sequence approximately in half, and the second step is to again divide the half sequences into quarters. The small sequence segments are then aligned to each other at the same time. Once the small segments are aligned, they are reconstituted into their original sequences. The instantaneous alignment of the short segments to each other removes the problem of greediness.[98] Image adapted from Stoye.[105]

originate from mutagenesis studies or errors in the expression of proteins used to obtain 3-D structures. The Sequence to Coordinates (S2C) website[106] is a database allowing one to enter the PDB code of a desired template protein. The differences between the sequence of the 3-D structure and the accepted amino acid sequence are returned in a plain text format containing additional

information. At the end of the file is the mutation record, which indicates the type and location of the mutated amino acid residue present in the PDB file and the correct amino acid residue according to the ExPASy protein sequence record. The additional information includes the SEQRES (the amino acid residues of the protein) and ATOM (the atomic coordinates portion of the PDB file) three-letter residue codes, the ATOM residue number, and the secondary structure based on the PDB file along with the STRIDE[73] method used for determining that secondary structure. Other useful information includes the method that was used for determining the 3-D structure, the structure's resolution, *R*-factor (how well the refined X-ray structure matches the experimental data), *B*-factor (a measure of isotropic variance), and the sequence file identifier that was compared with the PDB file.[107]

The method that scores (evaluates) the quality of the alignment of two sequences considers the number of gaps that were introduced into the sequences to aid the alignment; more gaps generally lower the overall alignment score. It is for this reason that one should know what segments of the sequence are responsible for a specific function. Having knowledge of the domain of the sequence (the bioactive portion of the complete amino acid sequence that excludes the pre- and proprotein portion of the sequence) is thus useful, especially if aligning distantly related sequences. Regions of the sequence that are precursors (used for signaling) and domains (chains) can be found in the Features section of a Swiss-Prot[28,51] entry. The Features section also includes information about mutated residues and the corresponding reference(s). Structural information about disulfide bonding, metal ion binding (and its purpose), secondary structure (known or predicted), and ligand binding sites are also provided.

When a protein's sequence is determined from the gene sequence of the ribosome, all of that protein's amino acid residues are included. This sequence is called the preproprotein (prepeptide), and most proteins are synthesized and transported from the cell in this form. Typically the protein is synthesized by the ribosome as a preproprotein. The prepeptide portion is for signaling, and the propeptide portion keeps the protein biologically dormant. The signaling segment is removed as the proprotein is moved into "storage" for use at a later time. When the active form of the protein is needed, it is removed from storage and the propeptide is cleaved. The propeptide segment occurs either at the beginning or the end of a protein's sequence. Databases containing the solved 3-D structures of proteins typically do not contain the pre- and propeptide segments because preproproteins and proproteins usually exist naturally in only small quantities.[16] It is imperative, therefore, that the alignment includes only the relevant portions of the sequences that best mimic the final, bioactive sequence of the protein of interest. Including pre- and propeptide segments can lead to incorrectly determined alignments and structures, which results in a useless protein model. The tools provided in Swiss-Prot can help the modeler determine what segments are important.

Alignment Basics

Although alignment programs differ in the methods actually used to align the sequences and how they score the final alignment, they share several common features, including gap penalties, similarity matrices, and alignment scores. Gap penalties reduce the quality (score) of an alignment when frequent "breaks" in the sequence are created. Similarity matrices indicate the probability of an amino acid residue's propensity to mutation over many generations (evolutionary distance). Alignment scores assess which of the many possible alignments is optimal.

Even though the alignment score is the last thing to be calculated, we discuss it first to explain why gap penalties and similarity matrices are important. Alignment scores are not universal; they can vary depending on the alignment method used and the gap penalty values implemented. A perfect multiple sequence alignment sometimes requires the introduction of gaps into the protein sequences during the alignment phase. Adding a gap (or two) ensures that the same or similar type (physicochemical, polar, charged, or shape) of amino acid residue is aligned, which in turn increases the score of an alignment. A gap penalty is assessed each time a gap is initiated or extended, and different penalty values can be assessed for starting a gap versus extending an existing gap.

The pairwise alignment of a target sequence to a single template is usually a straightforward task to accomplish. When several templates are needed for the modeling of a protein, the task becomes nontrivial because a multiple sequence alignment is required. When the target-template(s) sequence identity is above 40%, satisfactory results can be obtained with automatic sequence alignment methods,[108] but, when below 40% with respect to the global alignment, gaps and ambiguities are introduced. Because the 3-D, tertiary structure of homologous proteins is more conserved in evolution than is the corresponding primary structure,[109] a good strategy is to include 3-D structural information in the alignment process.[110] When superimposing the template structures, it is easy to distinguish regions of high and low conservation. Regions of high conservation are commonly referred to as structurally conserved regions (SCRs) and correspond to secondary structural elements such as α-helices and β-strands, whereas regions of low structural conservation are called structurally variable regions (SVRs) that are associated with loops or random coil regions. The optimal alignment reflects sequence conservation at the SCR and variability at the SVR, and several computational strategies exist that can improve a multiple sequence alignment.

A structure-based alignment of multiple templates is more accurate than sequence-based alignments because of the emphasis placed on aligning defined secondary structure regions (α-helices and β-sheets). The structural alignment highlights evolutionarily conserved residues that are part of defined secondary structures, which thus provides a template sequence with additional secondary

structure information. Additionally, if a large evolutionary distance exists between the target sequence and the templates, including sequences with different degrees of divergence, forming a bridge between the templates and the target can improve the alignment.[98] Only sequences that can be aligned to the templates reliably should be added. The same can be done with the target sequence. Sequences that share sufficient homology with the target sequence are multiply aligned. A profile (a numerical representation of the sequence that encodes information about its physical features) is calculated for each alignment, and the two profiles are then aligned with each other.[6] Careful examination of the obtained alignment followed by manual editing is usually needed to obtain a reasonable level of accuracy, especially if the target-template sequence identity is low. Placement of gaps should be avoided in secondary structure elements, in buried regions, or between residues that are far apart in space.[6]

Similarity Matrices

Aligning sequences by hand is a time-consuming and tedious procedure, so automated methods are the best way to align sequences in a fast and consistent manner. A method is needed to align the sequences based on their identity, physicochemical properties, or substitutions observed in nature, or based on residues with similar genetic codons. Similarity matrices determine the probability of mutation for a specific amino acid residue type. They are thus useful in the alignment of protein sequences. The similarity (scoring) matrices are of dimension 20×20 (amino acid residue versus amino acid residue). Upon diagonalization, similarity scores are determined between all amino acid residues. Amino acid residues with high physicochemical similarity, and that are commonly mutated because of genetic malfunctions or that are substituted across species, are given higher mutation probability scores. Each similarity matrix contains a set of values that are proportional to the probability of amino acid i having been mutated to amino acid j for a given evolutionary distance. Three common scoring matrices exist: Point-Accepted Mutation per 100 amino acid residues[101–103] (PAM) matrices, BLOck SUbstitution Matrix[104] (BLOSUM) matrices, and Gonnet[111,112] matrices. The PAM matrices are based on mutations observed from *global* alignments of closely related sequences that include both conserved and nonconserved regions. In contrast the BLOSUM matrices are based on blocks of local similarities and are the recommended matrix type for BLAST searches.

The PAM matrices,[101–103] which were developed by Dayhoff et al., are based on the probability of an amino acid residue mutating to another amino acid residue. The original PAM matrix was derived using a small group of closely related sequences and tracking the amino acid residue substitutions.[1] Each evolutionary PAM matrix is determined by multiplying the original PAM matrix by itself $n - 1$ times, where n is the number of desired evolutionary

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	2	-2	0	0	-3	1	-1	-1	-1	-2	-1	0	1	0	-2	1	1	0	-6	-3
C		12	-5	-5	-4	-3	-3	-2	-5	-6	-5	-4	-3	-5	-4	0	-2	-2	-8	0
D			4	3	-6	1	1	-2	0	-4	-3	2	-1	2	-1	0	0	-2	-7	-4
E				4	-5	0	1	-2	0	-3	-2	1	-1	2	-1	0	0	-2	-7	-4
F					9	-5	-2	1	-5	2	0	-3	-5	-5	-4	-3	-3	-1	0	7
G						5	-2	-3	-2	-4	-3	0	0	-1	-3	1	0	-1	-7	-5
H							6	-2	0	-2	-2	2	0	3	2	-1	-1	-2	-3	0
I								5	-2	2	2	-2	-2	-2	-2	-1	0	4	-5	-1
K									5	-3	0	1	-1	1	3	0	0	-2	-3	-4
L										6	4	-3	-3	-2	-3	-3	-2	2	-2	-1
M											6	-2	-2	-1	0	-2	-1	2	-4	-2
N												2	0	1	0	1	0	-2	-4	-2
P													6	0	0	1	0	-1	-6	-5
Q														4	1	-1	-1	-2	-5	-4
R															6	0	-1	-2	2	-4
S																2	1	-1	-2	-3
T																	3	0	-5	-3
V																		4	-6	-2
W																			17	0
Y																				10

Figure 8 Depicted is a common representation of a PAM similarity matrix after 250 PAM cycles. BLOSUM and Gonnet similarity matrices are also represented in this manner.

cycles. The basic premise of a PAM matrix is that the amino acid residue in column *i* will mutate to the amino acid residue in row *j* after the number of evolutionary cycles. The original One Point Accepted Mutation (PAM1) defines a unit of evolution corresponding to 1-point mutation for every 100 residues; a single PAM cycle is equivalent to approximately 1% of the residues mutating in a given sequence. The PAM family of matrices was developed by extension of PAM1. A typical example of a PAM matrix is PAM250 (Figure 8), which denotes that the matrix is based on the probability of an amino acid residue mutating to a specific amino acid residue after 250 point-accepted mutations. Because mutations can occur many times at the same site over evolutionary time, the PAM250 matrix corresponds to approximately 20% of the original residue types remaining at their initial location.[1] Positive values indicate that mutation to the corresponding amino acid residue type is likely, whereas negative values indicate that a mutation is improbable. An important issue to be aware of is that the amino acid residue of interest can mutate and then return to its original state, or it may not mutate at all. The similarity of the sequences to be aligned dictates which PAM matrices should be used; sequences that are very similar in composition should use PAM matrices with low numbers, whereas sequences that are related over large evolutionary distance should use PAM matrices with a large value. Figure 9 provides insight into the probability of specific amino acid residue types mutating in the PAM methodology.

By examining the PAM20 matrix we see that there is a small probability (1) that a mutation to or from tyrosine (Y) to a phenylalanine (F) will happen, but almost no chance (−19) of tryptophan (W) mutating to glutamic acid (E). It is reasonable to expect that a conversion between a tyrosine and phenylala-

nine will occur because the only difference between the two amino acid residues is a hydroxide group in the *para* position of the benzene ring (Figure 10). In the PAM100 matrix, the probability of mutation is increased for all residues, whereas the likelihood of remaining the same residue type is reduced. Specifically, the probability of mutating to a tyrosine from a phenylalanine is increased (from 1 to 4), as is the probability of tryptophan's mutation to glutamic acid (from −19 to −9). After more PAM cycles, the PAM300 similarity matrix shows that the probability of a tyrosine–phenylalanine mutation has again increased (from 4 to 9), as has the probability of mutating from

PAM20

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	6	-8	-4	-3	-9	-3	-8	-6	-8	-7	-6	-5	-2	-5	-8	-1	-1	-3	-16	-9
C		10	-16	-16	-15	-11	-8	-7	-16	-17	-16	-13	-9	-16	-9	-4	-9	-7	-18	-5
D			8	2	-17	-4	-5	-9	-6	-15	-13	1	-9	-4	-12	-5	-6	-9	-17	-13
E				8	-16	-5	-6	-6	-5	-10	-8	-3	-7	0	-11	-5	-7	-8	-19	-9
F					9	-10	-7	-3	-16	-4	-5	-10	-11	-15	-10	-7	-10	-9	-6	1
G						7	-10	-13	-8	-12	-10	-4	-7	-8	-11	-3	-7	-7	-17	-16
H							9	-11	-8	-7	-13	-1	-5	0	-3	-7	-8	-7	-8	-4
I								9	-7	-2	-2	-6	-10	-9	-6	-8	-3	1	-16	-7
K									7	-9	-3	-2	-8	-4	-1	-5	-4	-10	-14	-10
L										7	0	-8	-8	-6	-10	-9	-8	-3	-7	-8
M											11	-11	-9	-5	-5	-6	-5	-2	-15	-13
N												8	-7	-5	-7	-1	-3	-9	-9	-5
P													8	-4	-5	-3	-5	-7	-16	-16
Q														9	-2	-6	-7	-8	-15	-14
R															9	-4	-8	-9	-3	-11
S																7	0	-8	-6	-8
T																	7	-4	-15	-7
V																		7	-18	-8
W																			13	-6
Y																				10

PAM100

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	4	-3	-1	0	-5	1	-3	-2	-3	-3	-2	-1	1	-2	-3	1	1	0	-7	-4
C		9	-7	-8	-7	-5	-4	-3	-8	-8	-7	-5	-4	-8	-5	-1	-4	-3	-9	-1
D			5	4	-8	-1	-1	-4	-1	-6	-5	3	-3	0	-4	-1	-2	-4	-9	-6
E				5	-8	-1	-1	-3	-1	-5	-4	1	-2	2	-3	-1	-2	-3	-9	-5
F					8	-6	-3	0	-7	0	-1	-5	-6	-7	-6	-4	-5	-3	-1	4
G						5	-4	-5	-3	-6	-4	-1	-2	-3	-5	0	-2	-3	-9	-7
H							7	-4	-2	-3	-4	2	-1	3	1	-2	-3	-3	-4	-1
I								6	-3	1	1	-3	-4	-4	-3	-3	0	3	-7	-3
K									5	-4	0	1	-3	0	2	-1	-1	-4	-6	-6
L										6	3	-4	-4	-2	-5	-4	-3	0	-3	-3
M											9	-4	-4	-2	-1	-3	-1	1	-6	-5
N												5	-2	-1	-2	1	0	-3	-5	-2
P													7	-1	-1	0	-1	-3	-7	-7
Q														6	1	-2	-2	-3	-7	-6
R															7	-1	-3	-4	1	-6
S																4	2	-2	-3	-4
T																	5	0	-7	-4
V																		5	-9	-4
W																			12	-2
Y																				9

Figure 9 These similarity matrices represent 20, 100, and 300 PAM evolutions. A positive value indicates that a mutation can happen, and a negative value signifies that mutation is unlikely. The larger the numerical value, the more likely the mutation will or will not occur.

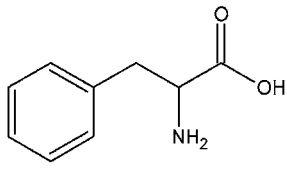
PAM300

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	2	-2	0	0	-4	2	-1	0	-1	-2	-1	0	1	0	-1	1	1	0	-6	-4
C		15	-6	-6	-5	-4	-4	-3	-6	-7	-6	-4	-3	-6	-4	0	-2	-2	-9	1
D			4	4	-6	1	1	-2	0	-4	-3	2	-1	2	-1	0	0	-2	-7	-5
E				4	-6	0	1	-2	0	-4	-2	2	0	3	-1	0	0	-2	-8	-5
F					11	-5	-2	1	-6	3	1	-4	-5	-5	-5	-4	-3	-1	1	9
G						5	-2	-3	-2	-4	-3	1	0	-1	-2	1	0	-1	-8	-6
H							7	-2	0	-2	-2	2	0	3	2	-1	-1	-2	-3	0
I								5	-2	3	3	-2	-2	-2	-2	-1	0	4	-6	-1
K									5	-3	0	1	-1	1	4	0	0	-2	-4	-5
L										7	4	-3	-3	-2	-3	-3	-2	2	-2	0
M											6	-2	-2	-1	0	-2	-1	2	-5	-2
N												2	0	1	0	1	0	-2	-5	-2
P													6	0	0	1	1	-1	-6	-5
Q														4	2	0	-1	-2	-5	-4
R															7	0	-1	-3	3	-5
S																1	1	-1	-3	-3
T																	2	0	-6	-3
V																		5	-7	-3
W																			22	0
Y																				12

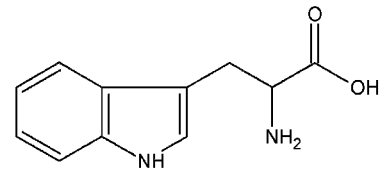
Figure 9 (*Continued*)

tryptophan to glutamic acid (from −9 to −8). Through the evolution of the PAM matrices, the probability of phenylalanine mutating to tryptophan (or vise- versa) improves (−6 to −1 to 1).

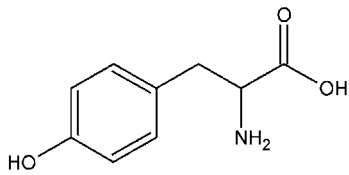
BLOSUM[104] matrices were constructed in a similar fashion to the PAM similarity matrices, but from a more diverse set of sequences. It is from this



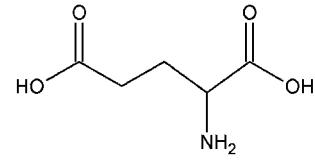
Tyrosine (Y)



Tryptophan (W)



Phenylalanine (F)



Glutamic Acid (E)

Figure 10 Amino acid residues. The obvious structural similarities between tyrosine and phenylalanine make it is easy to imagine the possible interconversion (mutation) between these two residues. The dissimilar structural features between tryptophan and glutamic acid illustrate the improbability of mutation.

standpoint that BLOSUM similarity matrices are considered more robust for the purposes of aligning amino acid sequences, especially those with low similarity scores (distantly related sequences).[113] The Gonnet[112] similarity matrices were developed around the same time as were the BLOSUM matrices and used the entire protein sequence database available in 1992. The Gonnet similarity matrix was constructed using the Needleman–Wunsch algorithm,[114] with indexing and reorganizing the amino acid sequences using a Patricia tree[111] on a small cluster of computers.

The alignment programs discussed below use similarity matrices—selecting the correct matrix for the desired set of sequences is important. The Gonnet similarity matrices are the preferred similarity matrices to use when aligning amino acid sequences because of the robust methods used in their development. Complicating the selection of the optimal similarity matrix is not knowing the amount of dissimilarity between the sequences to be aligned.

Clustal

A popular sequence alignment program is Clustal.[98,115,116] It is frequently used because of its ability to align large numbers of sequences of varying similarity quickly and accurately, not to mention its portability to numerous computer platforms.[115,116] The most recent version of Clustal is ClustalX,[115,116] which is an upgraded version of ClustalW[98] with a graphical user interface. Clustal is an iterative, progressive alignment method. Three steps are carried out in a Clustal alignment. First, each sequence is aligned to all other sequences in the alignment set (one at a time) to determine a distance matrix (pairwise alignments). Second, a guide tree is constructed based on the distance matrix. Third, the sequences are progressively aligned, based on the branching order in the guide tree. Figure 11 has three different representations of the same guide (phylogenetic) tree for the evolutionary relationship between several different species' lysozyme enzyme.

In the first step, the initial pairwise alignments can be performed using a fast but approximate method or by using a slower but more accurate method. The fast method finds the best alignment of two sequences based on a score that is determined by the number of identical residues (between the two sequences) minus a fixed penalty for every gap added. The slow method uses dynamic programming to align the sequences. It implements gap penalties for inserting a gap or extending a gap during the alignment process in addition to scoring the alignment. The score (percent identity) for both alignment methods is based on the number of most accurately aligned residues divided by the number of amino acid residues in the sequence (positions occupied with gaps are not included). Dividing the percent identity values by 100 and subtracting from 1.0, the distance matrix values (number of differences per residue) are thus calculated with no correction for multiple substitutions. After the initial alignments are prepared, a guide tree is constructed.

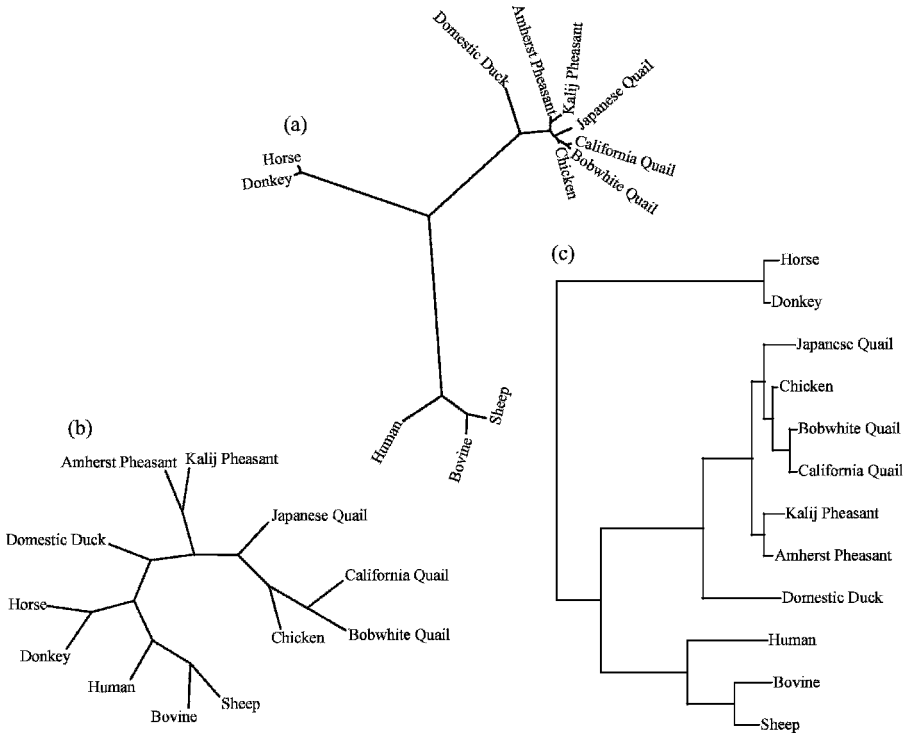


Figure 11 The evolutionary relationship between several species' lysozyme enzyme are displayed using three different guide tree (phylogenetic) representations. Progressive alignment methods use guide trees to aid in the alignment order of the sequences. The most related sequences are aligned first. Next, the second-most related sequence is aligned to the first two. This process continues until all sequences are aligned. These are unrooted phylogenetic trees representing the same information. The branches of trees (a) and (c) are proportional to the evolutionary distance, whereas the branch lengths in (b) are not. The guide tree relationship is based on values obtained from ClustalX,[115] and the images were created with PHYLIP.[117]

In the second step, a guide tree is used to aid in the final alignment of multiple sequences and is calculated from the distance matrix constructed in the first step. The guide tree is constructed by initially building an unrooted neighbor-joining tree (Figure 11) whose branches are proportional in length to the estimated evolutionary divergence of each sequence from the others. The assignment of a weight to each sequence is possible from the tree. The sequence weights are based on their distance from the root. In the case of sequences that share a portion of a branch (a common branch between several sequences), the weight of a single sequence can also be calculated (we use here human lysozyme as an example). The weighted value of such a sequence is the

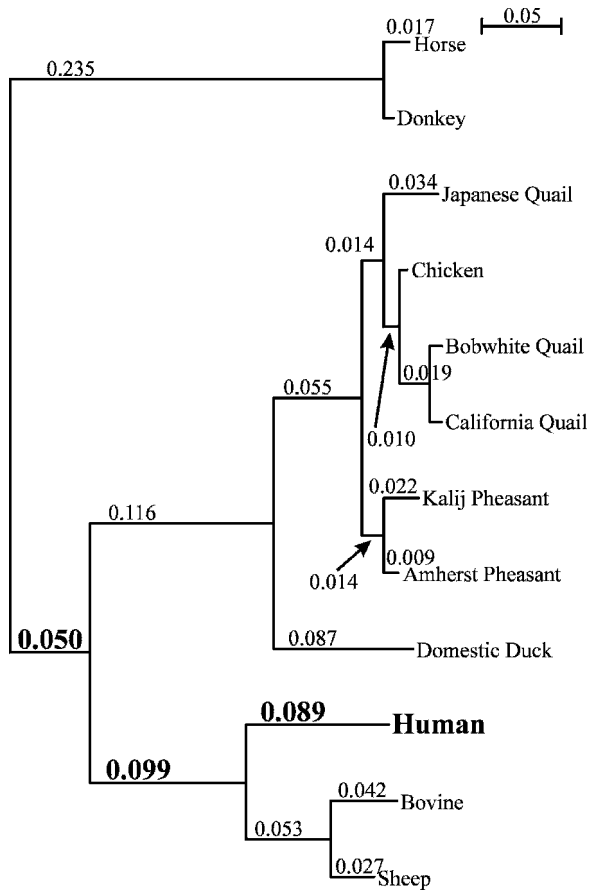


Figure 12 Guide tree with evolutionary distances. The values correspond to the branch length and thus the evolutionary distance (relationship) between the different proteins. The guide tree relationship is based on values obtained from ClustalX,[115] and the images were created with PHYLIP.[117]

sum of the length of the branches divided by the number of sequences sharing a specific branch. It is easiest to determine the weighted value by working from the tip of the branch to the root (Figure 12). For human lysozyme, it is accomplished by adding 0.089 to a third of the branch value shared with bovine and sheep (0.099/3) plus a tenth of the branch value (0.050/10) for the main branch. The weighting value for human lysozyme is then 0.127, and the total of all weighted values of all sequences in the guide tree equals 1. Further examination of the guide tree shows that some sequences do not have values, which indicates that their branch length is less than 0.001. Once the sequences are

weighted, they are normalized with the heaviest sequence being assigned a value of 1.0. The closely related sequences are assigned lower weighting values than are the divergent sequences because the former contain analogous information (similar protein sequences). All protein sequences are weighted the same when a *normal* (nonweighted) progressive alignment algorithm is used.

The third step is progressive alignment. This is a series of steps that aligns sequences pairwise to an ever-increasing group of sequences based on the guide tree constructed in the second step. The process of aligning the sequence starts at the tips of the branches and works toward the root. Sequences that share a common branch are aligned to each other first. Inserted gaps and extension of existing gaps permitted in previous alignment steps are retained and fixed. Progressive alignments are improved by weighting the sequences with normalized values based on the evolutionary (guide) tree. The weights are proportional to the number of closely related sequences that each protein sequence has; sequences that are closely related to many other protein sequences (redundant information) have a lower weight assigned compared with dissimilar sequences. The weights are used to determine the alignment order. Closely related sequences are aligned first to reduce the likelihood of following false trends that might be initiated by evolutionary-distant sequences. By delaying the alignment of the divergent sequences (i.e., those sequences with 40% or less identity) until all similar sequences have been aligned, one might locate the correct placement of gaps and be able to align the weakly conserved residues.

Another method of improving progressive alignments is to vary the gap opening and gap extension penalties. Using a strategy that is similar to how the sequences are weighted based on their similarity, the gap-opening penalty (GOP) is increased when aligning closely related sequences but reduced for distantly related sequences. The gap-extending penalty (GEP) is related to the difference in length between the sequences being aligned. Sequences with a large difference in length are assigned a larger GEP (to prevent large gaps in a short sequence) than those GEPs for sequences of similar length. In addition to imposing GOPs and GEPs, the gap penalty assessed in Clustal is position specific. Before sequence alignment, a GOP table is constructed for each pair of sequences to bias the initial GOP based on position. The gap penalties are implemented in a hierarchical fashion. To promote the insertion of a gap where gaps already exist, the GOP and GEP values are reduced and other gap penalty rules are waived. The GOP is increased for gap initiations within eight residues of an existing gap to discourage formation of adjacent gaps. Gap formation in hydrophilic regions (corresponding to loop or coil regions) is not heavily penalized when compared with opening or extending gaps in other regions. A hydrophilic region is considered to be a polypeptide segment containing five or more hydrophilic amino acid residues (D, E, G, K, N, Q, P, R, or S)[98] in a row. For such cases, the penalty for a gap insertion is reduced by

one third. The GOP for a nonhydrophilic region is subject to a predetermined set of values that are based on the probability of a gap occurrence adjacent to a particular residue. These probabilities are based on the alignments to known structures.[98]

Tree-Based Consistency Objective Function for Alignment Evaluation (T-Coffee)

T-Coffee[118] is an attempt to rectify a common problem in progressive-alignment (heuristic) methods in which errors committed in the first alignment cannot be corrected as other sequences are added to the alignment. T-Coffee like the other progressive methods can suffer from "greediness." Greediness refers to the inability of the alignment method to correct errors (addition or extension of a gap) made in the alignment process. The alignment of new sequences (information) might indicate previous errors in the alignment.[98] For example, in the alignment of six sequences, the alignment of the fifth sequence to the four previously aligned sequences indicates an improved alignment with the removal of a gap; yet the alignment of the sequences cannot be corrected to reflect the improvement. T-Coffee aims to reduce the probability of misalignment by using knowledge about all sequences at the beginning of the alignment process; more specifically, it uses the order of the sequences from the guide (phylogenetic) tree. There are five steps involved in a T-Coffee alignment: (1) Generate a primary alignment library, (2) derive the primary library weights, (3) combine the libraries, (4) extend the libraries, and (5) perform a progressive alignment.

The initial step of a T-Coffee multiple sequence alignment is to create a library by the pairwise alignment of all sequences of interest to determine their local and global alignments. The global alignments are determined using ClustalW[98] (version 1.75 with default parameters) to provide the "full-length," or slow alignment, as described earlier. The local alignments are performed using the Lalign[119] algorithm of FASTA.[120,121] It gives the 10 best nonintersecting local alignments that are stored in a library as pairwise residue matches. The pairwise residue matches are considered to be a constraint, but each match does not possess the same importance (weight) because some alignments (or segments of the alignments) are considered to be more correct than others. The second step involves calculation of the primary library weights using the aligned residues. The third step is to combine the local and global alignment results into a single, composite library. Weights are assigned to each pair of aligned residues in the combined library based on how often the pair of residues align with residues from the other sequences. The fourth step shapes the weighted constraints to fit a multiple alignment using library extension (a heuristic method). For any pair of residues in the library, a final weight is constructed that adequately illustrates a portion of

the information from the entire library, which is library extension.[118] Using the information gathered, step 5 proceeds using a progressive alignment strategy. The use of a phylogenetic tree (like those derived in ClustalW) is used to determine the sequence pairing for the multiple sequence alignment. The initial sequence alignment is the one with the most closely related sequences as noted by the phylogenetic tree. T-Coffee offers a method to create multiple sequence alignments using information from all possible sequence alignments along with an optimization method to derive the best multiple sequence alignment.

Divide-and-Conquer Alignment (DCA)

This method differs from the other alignment methods by aligning the sequences simultaneously. The DCA[105,122,123] method uses the multiple sequence alignment[124,125] (MSA) methodology. DCA is a data modifier that can use any method of multiple sequence alignment. It is an approximate alignment solution method that uses an unbiased starting point. In addition to developing a robust alignment method, the authors of DCA also were concerned with developing a method that is not computationally demanding.

Sequence alignment with DCA involves three main steps: (1) Divide the sequences into smaller fragments, (2) align the divided sequence segments, and (3) reconstruct the individual aligned sequences. The most difficult part of the three steps is the first, splitting the sequences. The division (slicing) of the sequences reduces the search space for locating the optimal alignment by reducing the number of possible configurations. The sequences are initially sliced at their (approximate) mid-points, and further division of those fragments is done in a similar way until the lengths of the subsequences reaches a predefined length. The authors of DCA suggest segment lengths between 40 and 100 amino acid residues for the alignment of several sequences and segment lengths of 20 to 40 residues for large sets of sequences (alignment of 5 or more sequences). The division of the sequences into smaller segments results in several small alignment problems with a finite number of possibilities instead of one alignment problem with an intractable solution. The second step is the alignment of the divided sequence segments using MSA, which thus adds to the strength of the DCA method. The gap penalty for an alignment in DCA is additive, and the same penalty is assigned to the alignment for gaps occurring at the beginning, the end, or in between amino acid residues. The final step is to recombine the segments to form their original proteins preserving the sequences' alignment and inserted gaps. DCA uses information about all sequences during the alignment process in contrast to the progressive alignment methods, which look at only a small subset of the sequences.

The DCA method of sequence alignment differs from progressive alignment methods such as Clustal because it does not use guide trees to determine the alignment order of the sequences. Instead short segments from all sequences are aligned at once providing a way to compare and align all sequences at the same time.

The sequence alignment methods discussed are just a sampling of the available methods. The overall goal of automated methodologies is to create alignments that are difficult to improve on using human intuition. Some methods can align closely related sequences accurately, but when the alignment involves distantly related sequences (less than 40% similarity), the alignment becomes problematic. The importance of the sequence alignment cannot be stressed enough; it is considered to be the most important step[6] in a comparative protein modeling study.

Two schools of thought exist for the alignment of a target to a template. The first is to use multiple sequences covering an evolutionary range, and the second is to use only a single template sequence. Researchers in the multiple sequence school recommend using sequences from different species with varying similarity (sequences with $\geq$40% similarity to each other). The construction of an optimal alignment using multiple sequences relies on the conservation of some important residues and the mutation of other important residues through evolution of the species. Typically, but not always, a residue mutates to another residue with a similar structure, function, or physicochemical property over time. All sequences are initially aligned with each other in a pairwise fashion to determine the final order in which they will be aligned, as discussed. The target-template alignment school views the introduction of additional sequences as a possible source of error for the alignment. It is thought that the additional sequences provide the possibility of a chance correlation being found for the initial or subsequent alignments. A plethora of sequence alignment results in a given study does not ensure that the optimum alignment has been found; thus, the opportunity exists for misaligned sequences. These nonoptimal alignments are often the result of the automated alignment method's lack of knowledge about the protein's tertiary structure and other information regarding the system of interest. These misalignments are easily corrected; methods used to improve the alignments are discussed in the next section.

Example: Aligning Sequences

The original bovine α-lactalbumin protein model was constructed before the availability of efficient sequence alignment programs and computational protein modeling applications. Instead of searching for similar sequences and structures using BLAST (because it did not exist), Browne et al.[3] considered only four different solved protein structures as templates. To explore the

Table 6 The Percentage of Sequence Identity and Sequence Similarity (numbers in parentheses) Between Bovine α-Lactalbumin and the Template of Interest

CLUSTAL X Alignments	[a]Template Group	[b]Four Templates	[c]Five Templates	[d]Pairwise
Horse Hemoglobin α	12 (28)	7 (25)	7 (26)	11 (28)
Horse Hemoglobin β	14 (30)	9 (26)	10 (27)	14 (30)
Hen's Egg White Lysozyme	35 (54)	37 (56)	36 (54)	37 (56)
Sperm Whale Myoglobin	13 (25)	6 (17)	6 (18)	12 (24)
[e]Human α-Lactalbumin	73 (87)	–	73 (87)	73 (87)
T-Coffee Alignments				
Horse Hemoglobin α	10 (26)	10 (22)	6 (22)	10 (27)
Horse Hemoglobin β	13 (27)	12 (24)	12 (28)	14 (33)
Hen's Egg White Lysozyme	33 (53)	35 (54)	35 (54)	35 (54)
Sperm Whale Myoglobin	13 (26)	8 (20)	7 (20)	13 (26)
[e]Human α-Lactalbumin	73 (87)	–	73 (87)	73 (87)

[a]The **Template Group** consists of the sequence for a specific protein from several different species.

[b]The **Four Templates** alignments consists of the four templates considered by Browne et al.

[c]The **Five Templates** alignments consists of the four templates considered by Browne et al. plus the sequence of the human α-lactalbumin.

[d]**Pairwise** alignments are the alignment between bovine α-lactalbumin and the proposed template.

[e]Human α-lactalbumin was not included in the alignment of the initially considered templates to preserve continuity between the results discussed here and those of Browne et al.

relationship between these four templates and bovine α-lactalbumin, we compiled a small group of sequences for each template. The proteins within each group were aligned to each other and to bovine α-lactalbumin with ClustalX[98,115] and T-Coffee.[118] The default ClustalW parameters including the Gonnet similarity matrix[111] were employed for all alignments (no parameter options are available for the T-Coffee method). The purpose of these template-specific alignments is to determine which of the templates is most similar to the lactalbumin target. Comparing our results from the alignment of multiple sequences of a small group of template structures with the target sequence, one finds (Table 6) comparable or better results (based on sequence identity and similarity) than for the alignment of the target to a single, proposed template sequence or to the pairwise alignment of template and target.

The authors of T-Coffee boast that it is more accurate than ClustalW for aligning sequences with less than 30% sequence identity albeit requiring more time. The overall improvements gained by T-Coffee compared with ClustalW are not significant in this case. The percent identity and similarity for the horse hemoglobins and the sperm whale myoglobin is very low. Accordingly, protein models constructed from any of these templates would result in terribly misleading structures. The percent identity and similarity between bovine

α-lactalbumin and the egg white lysozyme are marginally acceptable, so a template could be selected. In this case, the ability to improve on the work of Browne et al. is possible; the construction of bovine α-lactalbumin from human α-lactalbumin is possible given its greater sequence identity to bovine α-lactalbumin (73% identity and 87% similarity). Several alignments from Table 6 are presented in Figure 13.

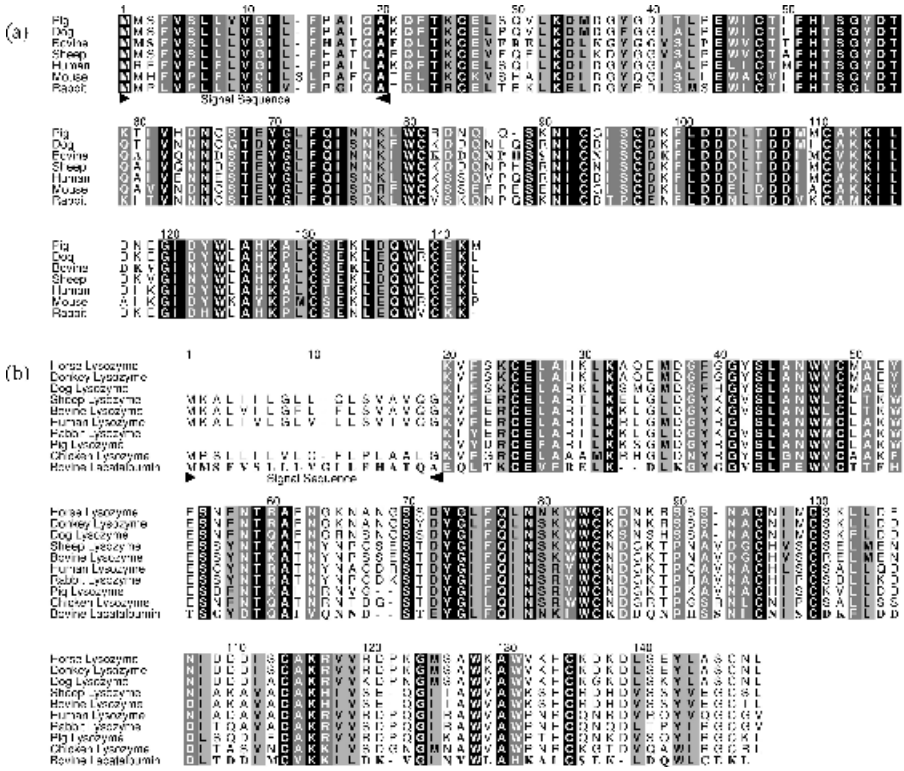


Figure 13 Bovine α-lactalbumin alignment to related sequences and possible templates. The alignments presented here range from the multiple alignment of sequences for the same protein (α-lactalbumin) from different species (a), to the multiple alignment of the target protein (sequence) to a collection of the same protein from different species of a probable template based on the work of Browne et al. (b) and considered templates (c), to a pairwise alignment of the target sequence (bovine α-lactalbumin) and a template (human α-lactalbumin) (d). The similarity of residues in a column is based on the work of Zvelebil et al.,[126] which calculates the residue similarity. Conserved residues have black backgrounds and white text, highly conserved residues have gray backgrounds and black text, and reasonably conserved residues have gray backgrounds and white text. The signal portions of the protein sequences are denoted with filled triangles. Images were created with JalView[127] and ALSCRIPT.[128]

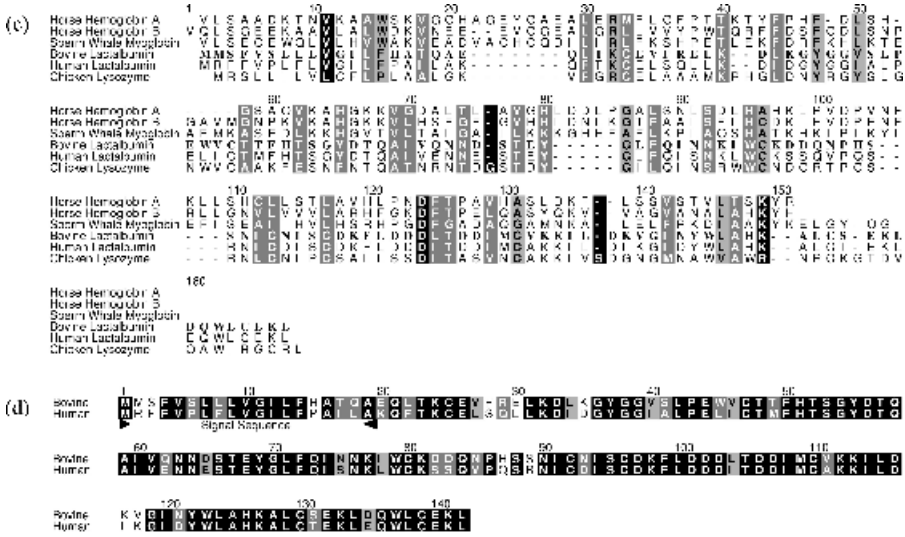


Figure 13 (*Continued*)

STEP 3: SELECTING TEMPLATES AND IMPROVING ALIGNMENTS

Seasoned professionals who work in the area of comparative modeling know that the order of business for step 3 is to first improve the alignment and then to select the template. This step of comparative modeling is a process in which the sequence of interest (the target) and the other sequences and structures (template) are aligned optimally. Once the sequences and structures are aligned, the best template(s) can be selected based on its evolutionary distance as determined by a polygenic tree (as discussed briefly). The condensation of these two steps into a united event draws attention to the fact that both steps can be considered to be small components of the overall process, but they can influence the outcome of the protein model dramatically. Combining these two steps might be considered to be an inconsistency with respect to the importance of each individual step, but by doing so, it allows for the smooth pedagogical transition from aligning the sequences and structures to the construction of the protein model.

Selecting Templates

When selecting the template structure for a protein model, it is important to consider the "quality" of the 3-D structure. It is done by considering the resolution, *R*-factor, and the anisotropy, *B*-factor. The *R*-factor and *B*-factor provide information about the refinement quality and variation (movement) of individual amino acid residues (and atoms), respectively. The resolution of a

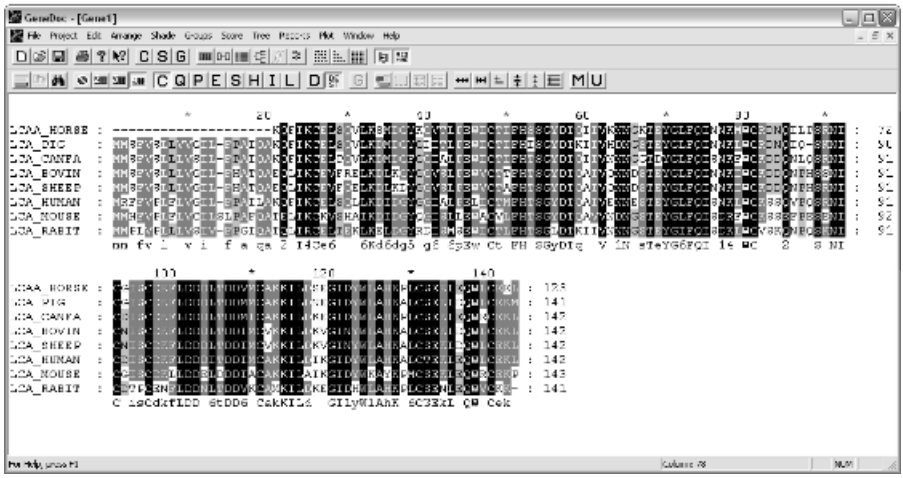
structure quantifies the lowest resolvable separation between two carbon atoms;[107] it is the ability to state that an atom occupies its specified portion of 3-D space (plus or minus the resolution value). The *R*-factor (the residual index) of a structure is the value that relates how well the refined structure matches the experimental results (electron density maps) and should not be considered as the "Holy Grail" for "correctness of fit."[107] The *B*-factor is a thermal measure of uncertainty (extent of electron density smearing) for the structure and is assigned to each atom and can be calculated for each amino acid residue. The largest thermal motions are typically observed in side chains and loops. It is not uncommon for several published X-ray versions of the same amino acid residue to be at different locations because of large *B*-factors. Atoms with *B*-factors greater than 50 $Å^2$ are considered to be imperfectly defined atoms, whereas those with *B*-factors less than 15 $Å^2$ are considered to be correctly defined;[129] varying levels of positional certainty exists between these two values. Atoms with *B*-factors approaching 100 $Å^2$ should be viewed as incorrect.

Selecting which template should be used for a protein model is, at times, trivial, especially if only a couple of related 3-D structures are available. In other cases, however, several protein structures are available from which to select (that are a similar evolutionary distance from the target sequence) and are either point mutations or have varying quality in crystallographic parameters, which thus makes the selection of a template difficult.

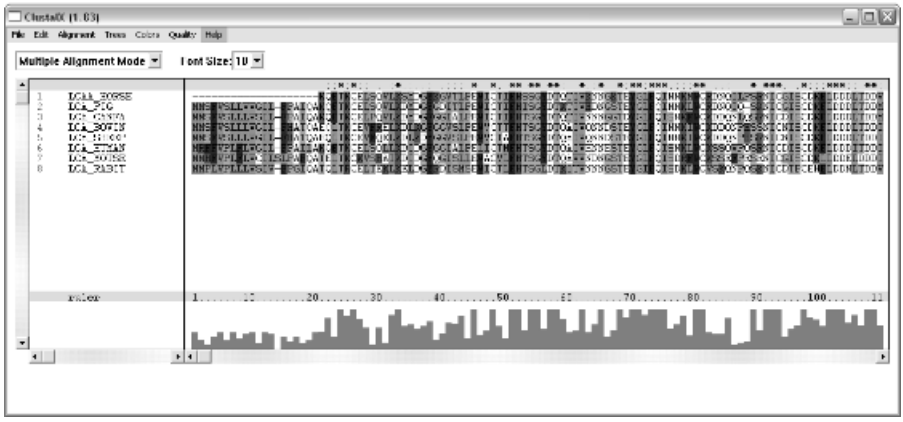
The automated alignment of sequences is desirable but is considered by many researchers to be inadequate because of the techniques implemented in the alignment methods. Some alignment methods discussed in the previous section were designed to eliminate the need for user intervention. However, even the best alignment of several protein sequences by automated methods does not guarantee a correct protein model will be generated because only the primary structure of the proteins is being considered with no knowledge of the secondary and the tertiary structure. The goal of aligning amino acid residue sequences is to match similar sections of amino acids (based either on amino acid residue properties or projected secondary structure features) with the anticipation that these regions will have matched structural features, such as secondary structures and catalytic residues. Without using scientific intuition, the automated processes could place a gap in the middle of an α-helix or β-sheet, something that does not happen in nature. Gaps are more suitably placed in regions of the template sequence lacking secondary structure such as in polypeptide loop (random coil) regions.

By using a hydropathy index (the probability of an amino acid residue being in a hydrophobic environment based on its neighboring residues in the primary structure) along with a predicted secondary structure, the target protein, in conjunction with the known secondary structure of the template structure, can be used to aid in the sequence alignment. The concept behind using hydropathy plots and a predicted secondary structure is to aid the alignment of the protein sequences. These topics are discussed later. Although the hydropathy

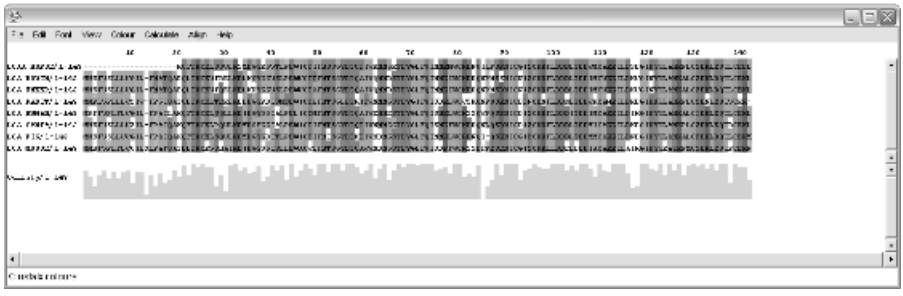
(a)



(b)



(c)



plots and secondary structure predictions rely on experimentally derived data, the predicted secondary structure could be correct or it could be completely wrong with respect to the actual secondary structure of the target protein. Moreover, the results can vary slightly or differ greatly when comparing the different methodologies; results that seem to be promising could be in fact very misleading. Sequences can be aligned manually using programs such as GeneDoc[130] or JalView[127] or aligned automatically with ClustalX[115,116] or JalView.[127] Screen images from these programs are shown in Figure 14. These programs display the alignment of multiple sequences and provide statistical information about the alignment. ClustalX does not allow one to manipulate individual amino acid residues. Instead it allows the user to evaluate the alignment and to select a segment of residues or several sequences to be realigned with the initial sequence alignment. In addition to providing numerical values to assist in the alignment, these programs display various shading methods based on physicochemical properties, conserved residues, identity to a specific sequence, and the overall property of the residues. The methods examined in this section are useful for improving the target-to-template sequence alignment. They can gather information about related sequences that do not have solved tertiary structure, and they provide a useful knowledge base built on a consensus of those acquired data.

Selecting the template (a known 3-D structure) for a comparative model is the most important step in the process of comparative (homology) modeling because selection of a wrong fold (protein motif) can lead to a wrong model more than can an incorrect alignment (this is not suggesting that sequence alignment is unimportant). The selection of the template should be based on a closely related protein within the same family as the target. Unfortunately, many proteins that one would like to model do not have templates that are closely related. In this scenario, the search for a suitable template must begin by seeking distantly related structures.

Improving Sequence Alignments With Primary and Secondary Structure Analysis

The primary structure of a protein contains information of use to homology modelers. A primary structure analysis can be used to predict α-helical or

Figure 14 Graphical sequence alignment programs. The ability to visualize the alignment of protein sequences and to manually adjust the alignment is important. GeneDoc (a) provides the ability to align sequences manually and provides statistical information regarding the alignment. ClustalX (b) provides the same functionality as ClustalW, but with the aid of a graphical interface. JalView (c) can be considered an improved version of ClustalX, providing various alignment methods (via the Web). In addition to aiding in the alignment of sequences, these three programs provide the ability to color-code alignments based on various properties, provide alignment statistics, and import and export files of various formats.

β-strands and coiled coil regions in proteins. A primary structure analysis can also reveal regions rich in proline, glutamic acid, serine, and threonine (PEST regions); locate protein sequence repeats; predict the percentage of buried versus accessible residues; and provide information about the system's isoelectric point. Significantly, the information stored in the primary structure is useful for determining the location of hydrophobic and hydrophilic regions of the protein. By using the amino acid residue sequence, a set of predetermined hydropathy values (a residue's affinity for water), and a window region, a hydropathic profile for a given protein can be determined. Hydropathy plots can be used to determine whether a protein contains any transmembrane regions or to locate regions (segments) of the protein that are buried (in a hydrophobic environment). The ability to predict the secondary structure of a protein helps the modeler to predict the 3-D (tertiary) structure of the protein of interest. Knowing what segments of the primary structure will form a specific secondary structure also helps the researcher classify the type of protein being studied. The prediction of a secondary structure makes use of the protein's sequence and is accomplished with either a database[131] to compare the sequence of interest with those with known structures or by an a priori prediction method.[70,71] The more common of the two approaches is to use a database to predict the secondary structure.

Hydropathy Plots

Hydropathy plots are used to predict buried and exposed regions of a protein. Use of these plots was first demonstrated by Rose[132] and Rose and Roy.[133] Hydropathy plots are based on Chothia's[134] observation that hydrophobic residues (amino acid residues with hydrophobic sidechains) tend to be buried when the protein exists in its native conformation. More than one set of hydropathy values is available. The best known hydropathy indexing methodology is that of Kyte and Doolittle.[135]

Calculating the hydropathy profile for a protein sequence is accomplished by determining the hydropathy score for each residue in a boxed region as one progressively moves down the biopolymer. The boxed region starts at the amino end of the sequence and moves toward the carboxyl terminus. The box size is of fixed dimension and is user determined. Kyte and Doolittle determined that an optimal box length is 7 to 11 residues.[135] Centering the box on the residue of interest and summing the hydropathy indices of all residues contained in the box provides a hydropathy score for that centered residue. This value is the hydropathy score of only the center residue even though it accounts for neighbor residues. Examples of hydropathy plots are depicted in Figure 15. Other parameters one can adjust include weighting the residues at the box edges, which invokes a weight variation for the model and normalizes the hydropathy indices.

There are two methods of varying the weight of the residues: linear or exponential. The default method of performing a hydropathy plot assumes that each residue is equally important, which thus produces a weight of 100%.

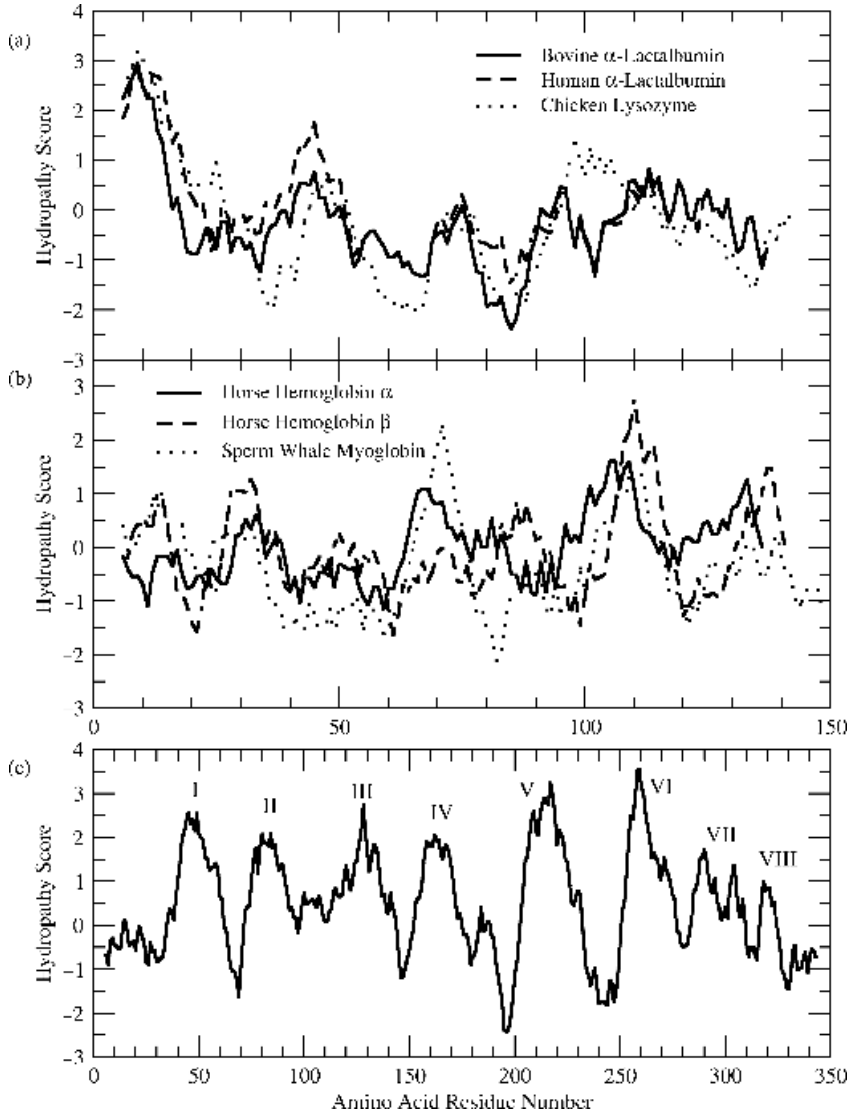


Figure 15 Examples of hydropathy plots. In (a), the hydropathy plots of bovine and human α-lactalbumin are very similar, whereas the hydropathy plot of chicken lysozyme differs but follows the general contours of the α-lactalbumin hydropathy profile. In (b), there is a general trend among the hydropathy profiles for horse hemoglobin α and β and sperm whale myoglobin. This is expected based on their similar structural features (all α-helical structure). Hydropathy plot (c) illustrates the approximate location of the seven transmembrane α-helices of bovine rhodopsin (I–VII), a transmembrane protein, and its intermembrane eighth α-helix (VIII). Hydropathy plots are best suited for providing insight into the physicochemical similarities between proteins and have the ability to elucidate the hydrophobic environment of a segment of residues. These hydropathy plots were constructing using the Kyte and Doolittle[135] hydropathy parameters and a residue window containing 11 residues, with edge residues retaining 75% of their original value compared with the center residue; the residues between the edge and center residues were linearly scaled (5% increments).

Regardless of the weighting method used or the weight assigned to the edge residues, the residue of interest always retains its full weight (100%). The residues at the edge of the box are assigned an edge value of 10% when the linear weight variation method is implemented. Thus, when the box size is set to 7 residues and the edge residues have a weight of 10%, the center residue (residue 4) retains its full weight, residues 3 and 5 retain 70%, residues 2 and 6 retain 40%, and residues 1 and 7 retain 10% of their original hydropathy indices.

Pattern and Motif-Based Secondary Structure Prediction

The use of pattern and motif-based methods to predict a protein's secondary structure involves matching the primary sequence of the protein of interest with solved 3-D protein structures. The computer programs that accomplish it are similar to the sequence alignment methods discussed, in which the sequence is read into the program and then compared with entries in a database. The database contains common structural motifs found in primary sequences. These prediction methods are therefore sensitive to the structures that constitute the database. Several well-known (and lesser known) methods for predicting the secondary structure of proteins have the ability to predict the secondary structure correctly 70% of the time.[136,137] The pattern and motif-based secondary structure prediction methods are a delicate balance between accuracy and robustness, and like artificial neural networks,[79] they can suffer from overgeneralizations. Well-known pattern and motif-based secondary structure prediction methods include PSIPRED,[93,94] GenTHREADER,[138] PREDATOR,[73,137,139] PROF,[140] MEMSAT,[141,142] and PHD.[143]

If more than one template was discovered during the template search stage, it is necessary to select the most suitable template (or templates) for the system at hand. A global pairwise alignment is the prime measure of the template quality. The template quality increases with the overall sequence similarity but is reduced with an increase in the number and length of gaps. For situations in which there is a small difference in the evolutionary distance between different templates, selecting "the most homologous template" is usually the best choice. For other situations, selecting several (two to five) templates to build an optimal 3-D model might be a viable option. For a multiple sequence alignment of the target with the templates, one can first create a phylogenetic tree[144,145] and then divide the sequences into subfamilies to help select the most suitable template. Other factors one might need to take into consideration when selecting templates include the resolution of the structure and the reliability of the template structure(s). The structure of a protein depends on the state in which it exists. These states may correspond to an opened or closed conformation, existing with or without a bound ligand, or being in a complex with another protein. The choice of the template thus depends on the state of the system one wishes to model.

Example: Aligning the Target to the Selected Template

Selection of a suitable template is very important; the better the sequence identity (and similarity), the better the protein model. Here we selected hen's egg white lysozyme as the template in contrast to the other candidates explored by Browne et al.,[3] Perutz,[146] and Perutz et al.[147,148] Our selection of this template is based on its sequence similarity and percent identity with the target. In addition to selecting lysozyme as a template, we also selected human α-lactalbumin as a template because of its high sequence identity with the target. (Table 6)

There are times when the alignment of the target sequence to the template requires additional adjustment to move a gap into a loop region. Aligning bovine to human α-lactalbumin in this example was straightforward and did not involve any gaps, so no alignment improvements were needed. The alignment of bovine α-lactalbumin to hen's egg white (chicken) lysozyme was more complicated, however, requiring slight improvement manually (using GeneDoc[130]) to preserve the trailing α-helical endcaps in helices 1 and 5. The improvement was made to the initial T-Coffee[118] alignment of bovine α-lactalbumin to hen's egg white lysozyme. Figure 16 illustrates the problems associated with a gap being located close to the beginning or the end of an α-helix. The original sequence alignment is presented in (a), where the gaps of interest are denoted with open arrows (⇧ and ⇩). The improved alignment is shown in (b), where one of the gaps was moved toward the center of the loops. Examination of the protein structures (template, T-Coffee model, and improved model) provides a visualization of the deleterious behavior caused by gaps placed near helices (and therefore other defined secondary structures of a template). The image (c) is an overlay of the three models with magnified views of the helical endcaps shown in (d) and (e). Notice the distortion imparted to the helices in the model created with the T-Coffee alignment. It is therefore imperative to place gaps in loop regions where they will have little influence on the defined secondary structure regions.

STEP 4: CONSTRUCTING PROTEIN MODELS

Three main families of algorithms exist for building the protein model. These families include (1) programs that use spatial restraints from the template structure(s) as a guide to build the target model,[149–152] (2) programs that fit short peptide segments of the target to the conserved main-chain atom positions of a structurally related template,[5,67] and (3) programs that overlay several templates from the same structural class (family) and use the best aligned segments of the target-to-template fit to construct a protein model.[43–48,153] In this section of the chapter, the methodology employed by several of the more common protein structure construction methods are described.

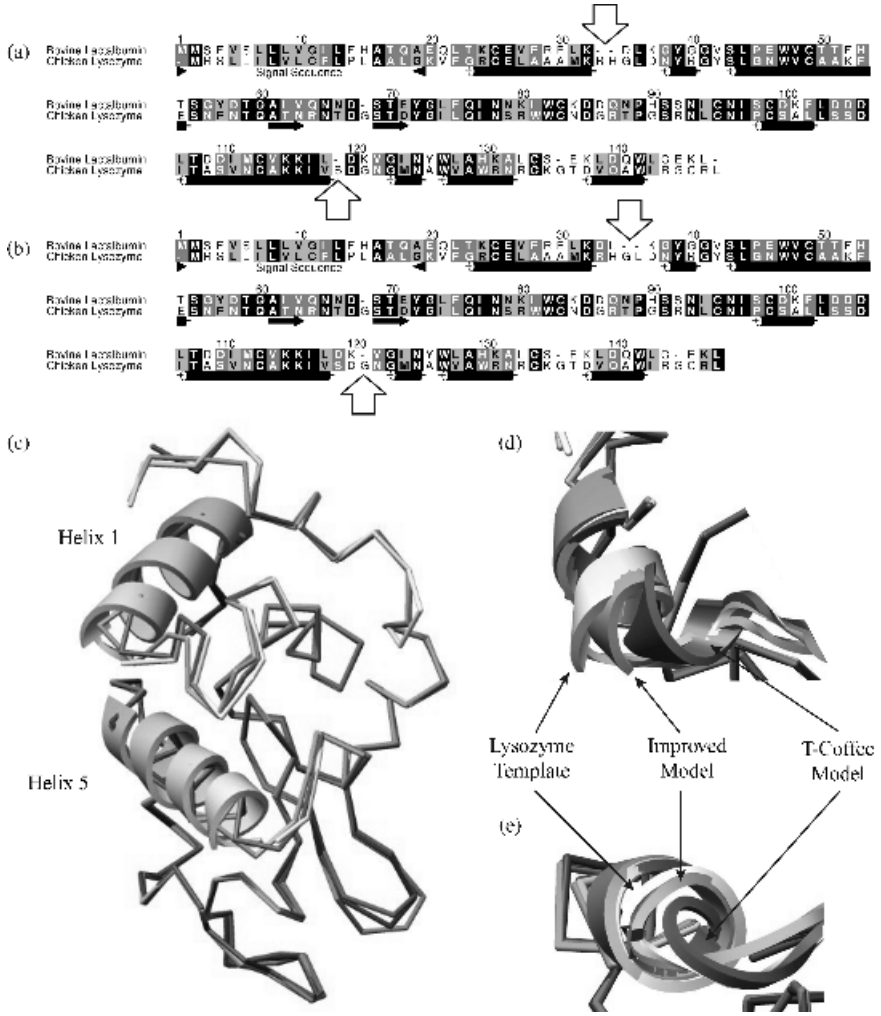


Figure 16 Improved alignment between bovine α-lactalbumin to hen egg white lysozyme. This small change in the location of gaps (alignments a and b) is important when trying to conserve the 3-D structure between the template and the to-be-built target models. The gaps of interest are indicated with open arrows (⇧ and ⇩). The initial alignment (a) was determined via T-Coffee, and the original gap locations were placed at the residue immediately after the α-helix termination site. The best location for gap placement is in loop regions, toward the center of the loop, because of the low sequence similarity of loops. The improved alignment (b) was obtained using GeneDoc. The ''tweaking'' or slight manual adjustment of the alignments resulted in an increase of sequence identity (35% to 36%) and sequence similarity (54% to 55%). This small gain in each score is very important. An overlay of the chicken lysozyme template, T-Coffee model, and the improved model is shown in (c). The regions of interest are Helices 1 and 5, which are illustrated as ribbon structures and enlarged in (d) and (e), respectively. Had the bovine α-lactalbumin models been created without user intervention, Helix 1 (d) would have extended beyond that of the template. The opposite is observed in (e), where Helix 5 of the T-Coffee model would have terminated prematurely compared with the template. Alignment images were created with JalView[127] and ALSCRIPT.[128] Protein structure images were created with UCSF Chimera.[99]

These programs include Satisfaction of Spatial Restraints[6,152,154] (SSR) as implemented in MODELLER,[155] Segment Match Modeling[5,67] (SMM) as implemented in MOE,[156] and the Multiple Template Method[43–48] (MTM) as implemented in 3D-JIGSAW[48,157,158] and in 3D-PSSM.[159–161] The SSR program of Šali and Blundell[7,152] constructs a 3-D protein model by using spatial restraints that are based on distances, angles, dihedral angles, pairs of dihedral angles, and other spatial features. It uses specific or pseudo-atoms derived from the template's structure. In contrast, is the SMM method of Levitt,[5] which searches a database of known protein structures to find similar sequences (related to the target) and structure (related to the template). The MTM method of Bates et al.[48,157,162] uses several templates and selects the segments with the best alignments for construction of the protein model. It is possible to use any combination of these methods to select templates, generate alignments, and refine or validate structures. The SSR, SMM, and MTM methods are described in more detail below.

Satisfaction of Spatial Restraints

MODELLER[6,152,155,163] is a comparative (homology) modeling package that uses Satisfaction of Spatial Restraints (SSR), a program that derives distance and dihedral angle restraints in the form of probability density functions from template proteins. MODELLER consists of a suite of applications that searches for and aligns a template structure(s) to the target sequence before constructing and refining the protein model. The spatial restraints methodology assumes that geometrical features, such as distances or angles, are conserved when comparing equivalent positions in homologous proteins. Therefore, constraints derived from the template(s) proteins can be used as a guide for the construction of the target model.[149,151,152,164] Havel and Snow[149] and later Srinivasan et al.[164] used distance-geometry techniques to create an ensemble of target structures that are consistent with lower and upper bound geometric constraints derived from the template(s).

Implementing the spatial restraints method consists of two main steps as described by Šali et al.[165] and Šali and Blundell.[152] The first step is to derive the spatial restraints based on the alignment. The second step involves construction of the 3-D protein model by fulfilling the spatial restraints of step 1. The first step relies on the pairwise alignment of the target sequence to the template structure. The alignment process of MODELLER is not unique and can be performed by other software. Using the alignment of the target to the template, the 3-D characteristics of the template are then projected onto the target sequence. The 3-D structure of the template together with its features that have been projected onto the target sequence comprise a knowledge base that is then used to construct the protein model. The rules used by SSR to construct a 3-D protein model are based on probability density functions (pdfs). The pdfs are a mathematical way of expressing the probability

of a certain physical (structural) feature occurring and are referred to as a restraint. The pdfs have the ability to take any form, but they must be nonzero and integrate to 1 over the full range of likely values for x.[152] The probability of existence for a structural feature (also called an event) is the result of

$$p(x_1 \leq x \leq x_2) = \int_{x_1}^{x_2} p(x)dx = 1 \qquad [2]$$

where x is an event that will take place between x_1 and x_2 (structure feature) and p is the probability. Each restraint (pdf) provides a distribution with upper and lower bounds for a given structural feature occurring as opposed to a mean value. Constructing a feature pdf from individual-basis pdfs is the first step in creating a molecular probability function. Features are properties linked to a single component (residue) of a protein or the association of properties between two or more components. Components can be singular, such as the percent sequence identity for a protein as a whole, or they can be pairwise such as the atomic distances between two residues as examples. Features can be defined as any measurement related to a set of specific atoms or residues, or, for that matter, even the entire protein. The feature pdf is a combination of all structural information that a specific structural feature can assess. The basis pdfs come from the template structure(s) and are joined into a single feature pdf using the alignment information. When using multiple templates, those same basis pdfs are weighted; the weighting is proportional to the difference of the average residue value and the specific residue of the template that has been aligned to the target residue.[152] The "final feature pdf" is constructed by multiplying the feature pdf by the van der Waals restraint. The feature pdfs are independent of each other, so the molecular pdf is a product of the individual feature pdfs.

When developing a molecular pdf, one must first determine the spatial restraints constituting a specific knowledge base of probability density functions. The spatial restraints are based on various structural relationships of the template and on the aligned target sequence. The pdfs provide a framework for assigning bond lengths, bond angles, and dihedral angles needed in the construction of protein models. The pdfs can be derived theoretically or empirically (from a collection of protein structures); in some ways, the pdf construction is similar to deriving a molecular force field where the bond lengths, bond angles, and dihedral angles are gathered and their average values and their associated standard deviations are determined. These stereochemical restraints are defined in the pdfs of SSR using classical molecular mechanics methods for the bond lengths, bond angles, and dihedral angles. The van der Waals repulsion term is modeled using its pdf term. The pdf for disulfide bonds is a generalization of the conformation of disulfide bridges as defined by Thornton[166] who analyzed disulfide bonds in solved, high-resolution X-ray structures. Unlike the theoretical pdfs that are known and are based on a myriad

of experimentally and theoretically determined molecular structures, empirical pdfs are derived solely from the structures of templates that have been aligned to the target sequence. Empirical pdfs are classified into three categories: (1) pdfs of individual amino acid residues, (2) pdfs for the overall protein, and (3) pdfs that account for the influence of the protein on an individual amino acid residue. The pdfs of independent amino acid residues take into consideration the type of residue involved (20 standard amino acids), secondary structure classification, the side-chain dihedral angle and its classification, the fractional solvent accessibility of the residue (of either the side chain, main chain, or entire residue), and the average *B*-factor (uncertainty) for side-chain atoms. These three general classes of pdfs are the building blocks on which the SSR methodology is built, and they are referred to as *basis pdfs*. The basis pdfs are converted to *feature pdfs* (the structural features of the template) to create the *molecular pdfs*, which are used to construct 3-D models of the target protein.

The next step in the SSR method is to build a model of the target protein by optimizing the molecular pdf. Optimization of the molecular pdf should provide a protein model with the most probable feature pdfs. The optimization of a molecular pdf is not solved in one step. Instead it is solved using the variable target function method[167] (VTFM) that breaks the molecular pdf into small and easy-to-solve segments. The VTFM then builds on these small segments until the complete molecular pdf is reconstructed. The VTFM initially starts to optimize the molecular pdf by using consecutive local restraints. It then adds intermediate restraints and ends with the addition of long-range restraints. The VTFM starts by determining the optimal conformation for one residue. The next step (and successive steps) uses the optimal conformation of the preceding step as its starting point. Successive conjugate gradient optimizations are performed by VTFM as each new residue is added. As the molecular pdf is refined and portions of the target protein model prove difficult to optimize, the restraints of the individual-basis pdfs are reduced by increasing the standard deviations (i.e., by allowing for less-than-optimal stereochemical configurations). To create an ensemble of target protein models, one starts the modeling of individual structures with different initial residues to provide different target conformations. Any violations in the target model that occur can be refined through the energy minimization function (molecular mechanics) in MODELLER, which employs a version of the CHARMM force field.[168] This geometry optimization persuades the protein model to conform to accepted stereochemical principles.

Segment Match Modeling

An assessment of different tertiary protein structures reveals that proteins with different biological functions often have different structural motifs. When dividing the protein structure into smaller segments, however, it is apparent that all proteins have similar substructures. Jones and Thirup[67]

developed this similarity concept when they solved the X-ray crystal structure of retinal binding protein, in which the authors noted a common similarity of turns between β-strands. Using short segments of three previously solved proteins, Jones and Thirup constructed a 3-D model of retinal binding protein. The short segments they used were based on the Cα carbons of known X-ray structures. The model they constructed from fragments had a root-mean-squared deviation (RMSD) of 0.95 Å (based on main-chain atoms) when compared with the experimentally refined coordinates of the retinal binding protein. This method was ground breaking in the arena of solving X-ray crystal structures, but it was time consuming and susceptible to bias because the fragments were overlaid onto the electron density maps manually.

The Segment Matching Modelling (SMM) algorithm developed by Levitt[5] automated both the search and the alignment of transferable protein fragments, thus removing bias and errors that could be attributed to human decisions. The SMM method was originally devised first to construct many protein models without user or database bias and second to provide an ensemble of credible protein models. Through randomization and averaging, the bias associated with intensive user input was removed. With SMM, the starting location for target protein building is selected randomly. A predetermined number of models is constructed and then averaged. The averaging provides information about the variance between the models; a small variance indicates regions of highly conserved structure (α-helices and β-sheets), and a large variance indicates variable regions (loops).

To construct protein models with SMM requires little change from the established steps of solving a protein's structure from X-ray scattering data. These steps are as follows: (1) Construct the database of segments, (2) build the target models, (3) randomize the construction of the models and construct the final model through averaging, and (4) minimize the energy of the final model. Step 2 actually consists of five individual steps, as we will discuss. The database of segments is composed of well-defined (low-resolution) protein structures and does not include duplicates or protein structures with minor modifications. A good selection of high-quality segments provides the modeler with a good chance of creating a sound protein model. As will be discussed, the intra-Cα distances match potential segments from the database to the template; these distances are calculated for all segments in the database. The "distance" described here is the number of amino acid residues separating the two Cα atoms; the range is from 2 to 19 residues.

Protein modeling with SMM needs the target's amino acid sequence and a template on which the protein model is based. It also needs an alignment of the target to the template (like other comparative modeling techniques). The process of constructing protein models with SMM proceeds in five steps: (1) Select a segment of the target protein to model, (2) construct a list of possible segments from the database that match the 3-D structure of the template, (3) sort the list of possible segments, (4) select the best segment, and (5) incorpo-

rate the coordinates from the best segments into the target protein structure. The first step involves a random selection of an amino acid residue. It is assigned to be the center residue of a segment consisting of an odd number of residues. When part of the template structure is missing (specifically three or more successive residues), as is the case when inserting amino acid residues, those segments are deferred in the modeling until the preceding, trailing, or both residues have been included. The hierarchy of missing atom reconstruction is to construct the missing main-chain atoms first and then to add side-chain atoms, as would be expected.

The second step of constructing a target structure is to compile a list of possible structural segments from the database of known protein structures that are based on sequence composition of the target and structure of the template. The segment searching consists of two subprocesses: constructing the main-chain and then appending the side chains. For construction of the backbone, it is not mandatory to have exactly the same amino acid residue type as in the target sequence at each segment's position. However, for modeling the sidechains, the specific amino acid residue is required, and the side chain of the residue should be positioned in a suitable conformation that is determined by the steric constraints of the environment (the currently modeled residues). The segment selected from the database for that initial target sequence must have parameters comparable with the template. The main-chain (backbone) conformation matching is based on the work of Jones and Thirup,[67] where segment conformation is based on intrachain Cα distances instead of direct Cartesian coordinate comparisons. Advantages and disadvantages exist when using the Cα distances for matching segments from a database to the template. The main benefit is that the matching is quick. However, it does not take into consideration the orientation (location in 3-D space) of the template's segment nor does it account for the chirality of the segment.[5] It is possible to find two segments with the same intra-Cα distances that are mirror images, which thus leads to two very different backbone conformations. It is possible to determine an approximate conformation of a polypeptide with Cα atoms, because Cα atoms do not dictate the conformation of small peptides.[5]

The third step of target construction checks the fit of the segments selected from the database to the target structure by calculating the RMSD and a van der Waals energy (the sum of the standard 6–12 Lennard–Jones interaction between atoms within 6 Å of each other) to evaluate steric interactions. The van der Waals' energy calculation includes different sets of atoms depending on the type of structure being evaluated. All atoms from the target structure are included except those in the first and last residues of the segment. When including atoms from the database segments into the van der Waals' energy calculations, two different sets of atoms are included; during the construction of the main-chain, only the Cα atoms of the segment need be included, whereas for modeling sidechains, all main-chain atoms and the entire residue of interest must be included. Calculating the RMSD and the

van der Waals interaction energy for each database segment being compared with the template is a time-consuming proposition, so this is reserved only for segments having the lowest inter-Cα distance deviation (the top segment candidates). The atoms used in the RMSD or the van der Waals' energetic calculations depend on the feature being modeled. As the target protein model is being constructed, the initially copied amino acid residues are unlikely to interact with each other. But as the number of residues increases, so does the likelihood of steric hindrance between adjacent side chains. As each top-ranked segment is fitted to the model, the RMSD and van der Waals' energy are calculated. The segment one would be inclined to include in the protein model is the one with the lowest RMSD or van der Waals interaction energy.

Selecting the best segment based on these two measurements can bias the segment selection, however. Instead, in the fourth step, the segment to be incorporated into the model is selected probabilistically from a subset of the low pseudo-energy group. A pseudo-energy is calculated for a small subset of segments from the database (usually those with promising inter-Cα distances when compared with the templates). The pseudo-energy is the sum of the RMSD value and one tenth the van der Waals' interaction energy for a specific segment. The fifth and final step of constructing a target structure is to copy the coordinates of the selected database segment and overlay them onto the template structure. To construct target residues where no template is available, the backbone is built in a way that the gap is grown together rather than from one end or the other. As the 3-D structure of the target protein grows, the added amino acid residues of the target structure can be used as reference points to construct the neighboring residue. The newly appended residues are then considered to be known and are then used for the selection and insertion of other residues. These five steps are repeated until the protein model is fully constructed.

Multiple Template Method

Both SSR and SMM usually rely on a single template structure to construct a protein model. In contrast, the Multiple Template Method[43–48] (MTM) employs several solved X-ray structures to construct a protein model of the target sequence. The multiple template concept is similar to using multiple sequences when devising the best alignment. With MTM, instead of using a single template, several protein structures from the same family are aligned (based on sequences and coordinates) and the template regions that are optimally aligned to the target sequence are used to construct the target protein model (Figure 17). Superposition of homologous protein structures reveals structural elements that are closely conserved. These structurally conserved regions (SCRs) are usually composed of secondary structural elements, the active site, and other essential structural residues. In between these conserved regions are structurally variable regions (SVRs), which may differ significantly

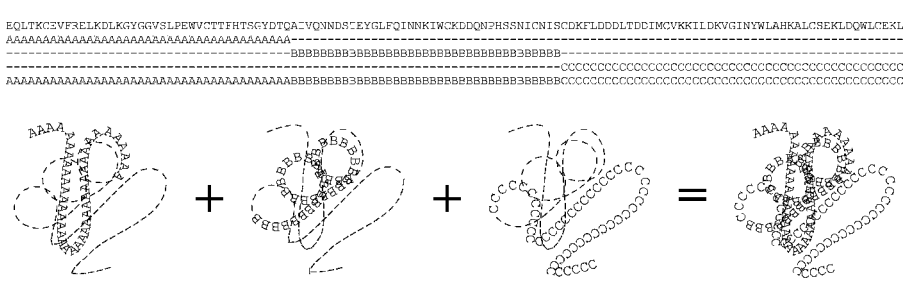


Figure 17 The ability to construct a target sequence from multiple templates is beneficial when several templates are available and when specific regions correspond to specific segments of the target protein. Three templates to construct the target protein are illustrated here. The initial template (AAAA) can be used to construct the N-terminus portion of the target protein, the second template (BBBB) for the middle, and the third template (CCCC) for the C-terminus. The three templates must be aligned with each other before constructing the target protein, thus providing the correct overall fold. The possibility of distorted stereochemistry at the merger points exists, but these imperfections can be removed via energy minimization.

in terms of their shape and composition even between members of the same family.[46] The variable regions are usually exposed loops at the surface of a protein.[45] The MTM methodology is similar to that of SMM in the search for and designation of specific regions (that are optimal) for the construction of the protein model from previously solved protein structures.

The multiple template methodology was developed independently by Chothia et al.[43] and by Blundell et al.[44] It is currently implemented in several comparative protein modeling packages including 3D-JIGSAW,[48,157,158] SAM-T02,[64–66] 3D-PSSM,[159–161] SWISS-MODEL,[42] MODELLER,[7] and MOE,[156] to name a few. The MTM requires significantly more user input than methods using a single template because of the need to select the regions (secondary structure) of greatest homology. We now explore the multiple template method 3D-JIGSAW for constructing the final protein model.

3D-JIGSAW

3D-JIGSAW[48,157,158] uses five steps to create a homology model: (1) selection and alignment of the templates (based on sequence and coordinates), (2) selection of the template segments, (3) creation of the backbone (framework, scaffold), (4) addition of the side chains, and (5) refinement and evaluation of the target protein model. 3D-JIGSAW is based on the database search of protein fragments developed by Jones and Thirup.[67] The decision about when to or when not to use multiple templates to construct a protein model is not simple and depends on the sequence identity between the target and templates as well as between the templates. Bates and Sternberg[48] delineate rules for when it is not prudent to construct protein models from multiple templates. They indicate that it is reasonable to use multiple templates when the

sequence identity between the target and templates is $\geq$40%, and the templates are very similar, but it is a bad idea to use multiple templates when the sequence identity between the target and templates is <40% and the templates are significantly different from each other. What these rules imply is that use of multiple templates that are similar or distant from each other will not yield better target models. If the templates are too similar, a single template should be used because the variation in secondary structure between the templates is smaller than that expected in creating the model. For templates that are too unrelated, it is advised to use the template that has the greatest sequence identity to the target sequence to reduce the modeling error. To construct good target protein models with multiple templates, it is best to have good homology between the templates and target sequences, while maintaining moderate similarities between the templates. This process ensures that the protein model is constructed from an ensemble of sequences having a greater homology to the target sequence than among themselves.

The initial set of sequences is selected as described earlier by carrying out BLAST searches. Those sequences are then aligned using one of the abovementioned multiple sequence alignment methods. For example, the PDB[36] can be searched for known structures to generate the initial set of sequences, and those structures can then be aligned in 3-D space via a superposition method. The crux of MTM is that it selects structures having a significant sequence similarity (identity) with the target sequence rather than using an individual template that is based on the best sequence identity. The structures selected are usually part of the same structural family (class) as the target protein. The template alignments with the target are done by using the known secondary structure of the templates. We already pointed out that the secondary structure of a protein is better conserved than are the loop regions. As a result, MTM focuses on the secondary structure of the templates; i.e., the alignment of the templates is defined by their secondary structure. The percent sequence identity between the target and the templates' secondary structure regions are then ranked to draw attention to the highly homologous segments. The selected portions of the templates thus constitute the core of the protein model. At this stage, the loops connecting those core structures are absent. Before the loops are considered, the SCRs that construct the secondary structure regions of the protein backbone are identified and a scaffold is built by averaging over the Cα coordinates of those templates.[169] The loops are then constructed to connect the secondary structures, which is done by using structural information from the original set of templates in conjunction with protein segments from a database. That information is then used in database searches to locate loop segments having similar connectivity structures and sequence identities. The fragments exhibiting the best structural compatibility with the secondary structure-to-loop transition are selected.[170] A quick energy refinement method can then be used to remove imperfections of the backbone. Once the scaffold has been constructed, the side chains are added (using rotamer searches that

will be described later). The side chains are added sequentially. First the corresponding atoms of the template side chain and the side chain to be added to the target are used as a base. Any missing atoms are added next using knowledge-based rules, or an entirely new side chain is selected from a rotamer library.[171] Constructing a protein model in this manner may not be a straightforward task to accomplish because the residues involved in the connection of different segments may contain large van der Waal's repulsion energies that require adjustment of main-chain torsion angles (followed by a short energy minimization without any restrained atoms).

The multiple template method software can also be used for aligning amino acid residues based on secondary structure or for selecting the best template. We describe MTM here in the section on model building rather than in the section on alignment because of its collection of methods that are used to construct a protein model. The true power of MTM lies in its ability to use several protein structures within the same fold (class or family) to construct a protein model having a similar structure. It is important when constructing protein models for proteins with similar structure but with different function. The MTM is also effective when the overall sequence identity between the target and the templates is low [even though some specific regions (secondary structures) may have noticeably significant sequence identity]. The MTM can also be invoked in MODELLER[155] and MOE[156] for protein model construction.

Overall Protein Model Construction Methods

Two philosophically different methods are commonly used to construct the target molecule's 3-D structure. One method is to use the best model, and the other is to use a consensus model. In the best model method, a series of intermediate protein models are constructed and the best one is selected to be the starting point for refinement (as the name implies). The consensus model uses intermediate structures to construct an average model, and that average model is used as the starting point for the refinement of the structure (as will be discussed in the NMRClust section). The number of intermediate protein models and the extent of energy minimization for the final protein model is user definable.

Template modeling is performed after the target sequence has been aligned with one or more template sequences. At this stage, the 3-D structure of the target sequence is still unknown by the modeler with any degree of certainty. For any given series of proteins, one can generally find two distinctly different types of secondary structure: defined and random. The first type of structure (defined) corresponds to conserved regions in which residue segments have maximum similarity or identity. These sequences are often found in the interior of the protein and have a defined secondary structure. The second type of structure involves variable regions, i.e., those parts of the protein ensemble

having a large variation in either the number or the arrangement of residues. These regions are often polypeptide loops that connect existing secondary structures. After aligning the sequences, there are three main steps to be carried out in the consensus method of homology modeling. First, the backbone of the structurally conserved regions is constructed. Then, the loops are added to this core structure. Finally, the side chains are added, the protein is geometry optimized, and that final structure is then validated. The major factor determining the quality of the constructed models depends more on the template selection and the target-template(s) alignment than on the manner in which the protein model is actually constructed. When used correctly, the different methods result in protein models of comparable accuracy.[6]

The three protein model constructing methodologies discussed in this section use different conceptual means to construct a protein model. The refinement methodologies inherent to these methods were not discussed. In the section, Step 5: Refinement of Protein Models, the refinement of the raw output of these methods (the protein model) to further reduce steric clashes is described.

Example: Constructing a Protein Model

The construction of several bovine α-lactalbumin protein models from human α-lactalbumin, chicken lysozyme, horse hemoglobin β, and sperm whale myoglobin templates was done using MODELLER.[155] Setting up MODELLER for constructing the protein models is a challenge; converting the sequence alignment format into files that MODELLER can understand is at times difficult because it lacks a graphical interface. The alignment of the sequence is saved (or provided) from the alignment program of choice. Usually the alignment is available in the Clustal alignment format (having an ALN suffix), which is the de facto file format for alignment programs because of Clustal's ubiquitous nature. The Clustal alignment format can be written and read by many different sequence alignment programs. The alignment file was then imported into GeneDoc[130] for alignment improvement, as discussed earlier. The improved alignment was then exported as a PIR file and converted into MODELLER-compatible files using PERL scripts[172] provided by the Biomedical Initiative Group at the Pittsburgh Supercomputing Center. The sequence of the template in the alignment file must match the sequence of the PDB file, and the use of the S2C[106] website was a quick and easy way to accomplish this. Occasionally an amino acid residue will occupy more than one conformation in a solved crystal structure, so it is imperative to ensure that only one conformer is included in the template PDB file before modeling the protein. MODELLER requires three basic files: (1) the alignment file, (2) the template PDB file, and (3) the MODELLER input file. The first and third files are simple data files containing information about the target and template(s) sequence alignment and the atomic coordinates of the template (Figure 18). The MODELLER input file (commonly referred to as a TOP

(a)
```
CLUSTAL X (1.83) multiple sequence alignment


LCA_BOVIN      MMSFVSLLLVGILFHATQAEQLTKCEVFRELKDLKGYGGVSLPEWVCTTFHTSGYDTQAI
LCA_HUMAN      MRFFVPLFLVGILFPAILAKQFTKCELSQLLKDIDGYGGIALPELICTMFHTSGYDTQAI
               [illegible]

LCA_BOVIN      VQNNDSTEYGLFQINNKIWCKDDQNPHSSNICNISCDKFLDDDLTDDIMCVKKILDKVGI
LCA_HUMAN      VENNESTEYGLFQISNKLWCKSSQVPQSRNICDISCDKFLDDDITDDIMCAKKILDIKGI
               [illegible]

LCA_BOVIN      NYWLAHKALCSEKLDQWLCEKL
LCA_HUMAN      DYWLAHKALCTEKLEQWLCEKL
               :*********:***:*******
```

(b)
```
C; bovine lca aligned to human lca

>P1;1b9o
structureX:1b9o:.:.:.:.::::
KQFTKCELSQLLKDIDGYGGIALPELICTMFHTSGYDTQAIVENDESTEYGLFQIS
NKLWCKSSQVPQSRNICDISCDKFLDDDITDDIMCAKKILDIKGIDYWLAHKALCTEKLEQWLCEKL*

>P1;blca
sequence:blca:.:.:.:.::::
EQLTKCEVFRELKDLKGYGGVSLPEWVCTTFHTSGYDTQAIVQNNDSTEYGLFQIN
NKIWCKDDQNPHSSNICNISCDKFLDDDLTDDIMCVKKILDKVGINYWLAHKALCSEKLDQWLCEKL*
```

(c)
```
INCLUDE                              # Include the predefined TOP routines

SET OUTPUT_CONTROL = 1 0 0 1 0       # uncomment to produce a large log file
SET ALNFILE  = 'alignment.ali'       # alignment filename
SET KNOWNS   = '1b9o'                # codes of the templates
SET SEQUENCE = 'blca'                # code of the target
SET ATOM_FILES_DIRECTORY = './:../atom_files' # directories for input atom files
SET STARTING_MODEL= 1                # index of the first model
SET ENDING_MODEL  = 50               # index of the last model
                                     # (determines how many models to calculate)
SET DEVIATION = 4.0
SET MD_LEVEL = 'nothing'

SET TOPLIB   = '$(LIB)/top_allh.lib'

SET TOPOLOGY_MODEL  = 1
SET HYDROGEN_IO     = on

CALL ROUTINE = 'model'               # do homology modelling
```

Figure 18 MODELLER uses these files to construct protein models. The ClustalX alignment file (a) is imported into GeneDoc (and the alignment modified if needed) and exported as a PIR file for conversion into the MODELLER alignment file format. Beneath the aligned amino acid residues are "*", ":", ".", or a blank indicating the amount of similarity between the aligned residues. The "*" identifies conserved residues, ":" denotes high physicochemical similarity between the aligned residues, "." signifies a low physicochemical similarity, and a blank indicates no similarity. The standard MODELLER alignment file for the creation of bovine α-lactalbumin protein model from human α-lactalbumin is in (b). The template is denoted with structure-X:1b9o, and the target sequence is sequence:blca. The end of the sequences are signified with *, and the template sequence must match that of the template PDB file. Two versions of MODELLER input files are presented in (c) and (d). The model creation file in (c) is the old format (from MODELLER7v7[7]) and is referred to as a TOP file. The new version of the model creation file (d) is from MODELLER8v0,[155] based on the Python scripting language.

```
(d) # model building
    #

    from modeller.automodel import *    # Load the automodel class

    log.level(output=1, notes=0, warnings=0, errors=1, memory=0)
    env = environ(rand_seed=-12312)  # To get different models from another script

    # directories for input atom files
    env.io.atom_files_directory = './:../atom_files'

    a = allhmodel(env,
          alnfile  = 'alignment.ali',      # alignment filename
          knowns   = ('1b9o'),             # codes of the templates
          sequence = 'b1ca')               # code of the target

    a.starting_model= 1                 # index of the first model
    a.ending_model  = 50                # index of the last model
                                        # (determines how many models to calculate)
    a.deviation = 4.0                   # has to >0 if more than 1 model
    a.md_level = None

    a.make()                            # do homology modelling
```

Figure 18 (*Continued*)

file) contains information that directs the protein building process. It should be noted that as of version 8.0 of MODELLER, the TOP scripting language has been replaced with Python scripts to direct the protein modeling. This input file contains information such as the location of the template and alignment files, the number of models to construct and how to refine them, along with additional instructions. MODELLER can refine the protein structure, but its ability to do so is not as robust as when using standard molecular dynamics programs like AMBER,[173,174] CHARMM,[175,176] Tinker,[177] GROMOS,[40] or NAMD,[178] which will be discussed later. Most users of MODELLER thus use it only as a protein model creator, and they then refine the target structure with more sophisticated programs designed for this task. The selection of the "best protein model" for refinement is usually done with software that one could call "protein model evaluators." The programs Protein Structure Analysis[179] (ProSa) and Verify3D[180,181] (discussed later) are examples of such software. In our example, 50 protein models were created from each template. Then, using Verify3D, the best model was selected and compared with its respective template.

STEP 5: REFINEMENT OF PROTEIN MODELS

Once the proposed protein structure has been constructed, it usually needs to be refined. For models developed from very similar templates (greater than 85% similarity), the need to refine the structures will likely be minimal, however. The creation and refinement of a protein model is considered by expert modelers to be a single process, but here, for simplicity, we treat

each as a separate step. A protein with all of its amino acid residues in the optimal conformation is considered a global minimum energy conformation (GMEC). Protein structures solved using crystallographic data are typically close to the GMEC. In contrast, target protein structures often are not at the GMEC, especially if the target and template sequences are dissimilar. It is for this reason that rotamer searches, energy minimizations, and molecular dynamics techniques are used to refine protein models. Rotamer searches and energy minimizations via molecular mechanics are methods that reduce steric hindrance of amino acid residue side chains. Molecular dynamics involves kinetic and potential energies so one can explore side-chain movement (for sampling of conformers), secondary structure fluctuation, and regions of the protein that might exhibit movement. This movement of protein regions is most commonly observed in loop (non-structured) regions and is possible in hinged proteins. Using a rotamer search for side-chain geometries before and after the molecular dynamics simulations can help locate the GMEC.

In this section, we describe four commonly used refinement methods. The first method is Side Chains with Rotamer Library (SCWRL) developed by Canutescu et al.[182] The goal of SCWRL is to predict the most plausible conformation of the amino acid residue side chains. The energy refinement methods of molecular mechanics (MM), molecular dynamics (MD), and molecular dynamics with simulated annealing (MD-SA) are related by the fact that they all use force fields. The MM method is a simple energy minimization of the protein model's atomic positions. The MM method is often used to remove repulsive contacts between amino acid residues of the protein model. The MD method is used to simulate how the protein model will interact with itself and its environment (either in water or in a biological membrane). The MD-SA method is a simulated annealing technique where kinetic energy is progressively removed from a MD simulation to find minimum energy conformations; as such, it can be used as a tool for conformational analysis. Because of the general nature and procedural style of this section, an example will not be presented.

Side-Chains with Rotamer Library (SCWRL)

In addition to predicting the backbone conformation correctly, one also needs to predict the conformation of the attached amino acid residue side chains. The rotational conformations of these side chains are referred to as rotamers. The side-chain orientation of amino acid residues typically occupy one of several discrete conformations.[183–186] The maximum number of possible rotamers for a protein is

$$\text{Number of Protein Structures} = \prod_{i=1}^{\textit{all residues types}} \left(\text{Number of Rotamers}^{\text{Number of Residues}}\right)_i \quad [3]$$

The number of possible rotamer configurations for a small peptide of 10 amino acid residues is approximately 1.5 million structures. This value is for the same backbone configuration and was derived by setting the number of rotamers for each residue to be four. Locating all possible rotamers for a small protein (100 amino acid residues) becomes an intractable problem. The need exists, however, to correctly and quickly predict the orientation and conformation of an amino acid residue in the target protein model, and it is for this reason that SCWRL was developed. Other methods and side-chain conformational libraries exist for doing this including the methods of Koehl and Delarue,[187] Lovell et al.,[188] and Xiang and Honig.[189,190] The different conformations for each amino acid residue are already known from a stored dataset of conformers, but their interaction with each other is not. In a side-chain search, only the side chains of the amino acid residues are moved to predefined conformations. Although SCWRL could be viewed as a model building tool because it constructs the most probable orientation of side chains for the target structure, it could also be considered to be a refinement method because it adjusts the location of the side chains to provide an energetically better protein model. In this discussion, SCWRL is viewed as a refinement method. SCWRL consists of two components: the rotamer library and the search method.

The rotamer library[191] used by SCWRL is backbone-dependent. Such libraries are considered to be more accurate than a backbone-independent rotamer library. A backbone-dependent rotamer library is more robust because the side-chain conformations are based on the protein's backbone dihedral angles (Φ and Ψ).[191–194] Both the secondary structure and the sequence composition are used by SCWRL to determine the most likely conformation of a residue's side chain. Contrarily, the backbone-independent rotamer libraries do not take a secondary structure into consideration for determining the orientation of the side chains. Eight steps are done by SCWRL to determine the best (most likely) side-chain conformation: (1) Read the initial structure, (2) determine probable rotamers, (3) define disulfide bridges, (4) perform a dead-end elimination (described below), (5) construct a residue graph, (6) determine the rotamer clusters, (7) find the most probable side-chain conformers, and (8) output the final structure.

The first step is to read the backbone coordinates of the protein model and compute the Φ and Ψ dihedral angles. This step also allows for an alternative (new) sequence to be read-in, which in turn signifies which amino acid residues to omit in a rotamer search (typically those that are conserved), and ensures that a complete chain is being explored (no gaps). The second step is to generate all possible rotational isomers (rotamers) at each amino acid residue position. It is not necessary to replace all side chains with ensembles of rotamers; sometimes it is desirable to leave the original Cartesian coordinates for conserved residues or for those known to form metal ion complexes. These residues with preassigned geometries are treated as steric boundaries in the search for the other side chains conformations. An ensemble of allowed

conformers is constructed for each residue from the possible rotamers. The ensemble starts with the most likely rotamer and is expanded by adding the next most likely rotamer until the sum of the probabilities for that set of rotamers is at least 90% (based on the summation of the individual rotamer's probability) for each residue. To determine if any amino acid residues have the ability to interact with their immediate neighbors one residue away, the Cβ distances between those residues are calculated along with the greatest distance between any side-chain atom and its corresponding Cβ atom.[182] In the third step, SCWRL determines which of the cysteine amino acid residues participates in disulfide bonds. Those side chains are then held fixed during the conformation search. SCWRL3.0 determines disulfide bridges based on an empirical scoring function that evaluates all cysteine pairings in a given protein structure. The empirical function uses the distance between the proposed bonded sulfurs, bond and dihedral angles, and the total "self-energy" for the two cysteine residues that might form the disulfide bridge.[182] The self-energy term consists of three parts: (1) a backbone energy term (derived from backbone-dependent rotamer probabilities based on Φ and Ψ), (2) a fixed side-chain energy term (interaction energy between the fixed side chain and any other unfixed side chain), and (3) the articulation point rotamers' interaction energy (the energy of the biconnected component, which will be discussed, that is composed of the self-energies of the other rotamers, the interaction energy of these rotamers interacting with other rotamers, and the interaction energy between the rotamer of interest and the articulation point). A disulfide bond is considered to exist between two cysteines if the score is greater than a predetermined value.

The fourth step performed by SCWRL determines which side chains are eligible for rotamer searches. A dead-end elimination (DEE) with the "Goldstein criterion"[195] is an efficient method of reducing the many possible conformers. A DEE is an algorithm that can correctly determine the next step using only the information of the current state. It focuses on the pairwise interactions between subsystems,[195] such as side-chain interactions, and separates the interactions between subsystems into their individual components of the energy potential function. It enables DEEs to identify and eliminate rotamers that are not solutions to the GMEC in an iterative process.[196] The DEE method is a common way of reducing the combinatorial conformer problem and has been implemented in several rotamer search programs.[182,196–198] The Goldstein criterion for rotamer selection considers two rotamers for the same amino acid residue. One rotamer is preferred over the other if it has a lower interaction energy with the surrounding residues' rotamers even if the other rotamer has a high probability as found in the library.[195] When many possible rotamers exist for a single residue, they are discarded beginning with the high interaction energy rotamers and working downward in terms of energy. At completion of the DEE step, residues with one rotamer remaining from the list of possibilities are held fixed for the remainder of the rotamer search. The efficiency of the Goldstein criterion comes from the significant reduction

in the number of possible conformers that must be computed at a later time with, say, a force field.

After the DEE step, all amino acid residues having two or more possible rotamers are noted (set "active") and a graph is constructed with each of those residues as a vertex. The interaction energy between pairs of "active" amino acid residues is then calculated. Once an interaction (i.e., a pairwise energy value between two residues not equal to zero) is determined, an edge of the graph (link) between the two residues is constructed and the search is halted. Although an active residue may have more than one rotamer, only one of the several possible rotamers on one active residue needs to have an interaction with another possible rotamer of a different active residue to construct a link between the two vertices of the graph. A connection graph is thus constructed as the fifth step of the process. The sixth step is to carry out a depth-first search number (DFN) to resolve the set of biconnected components and to locate the articulation point of the graph.[182] A biconnected component is an amino acid residue that is connected to (interacts with) other residues in the graph, and the articulation point is a residue that connects together two or more clusters of amino acid residues, as illustrated in panel (a) of Figure 19. It

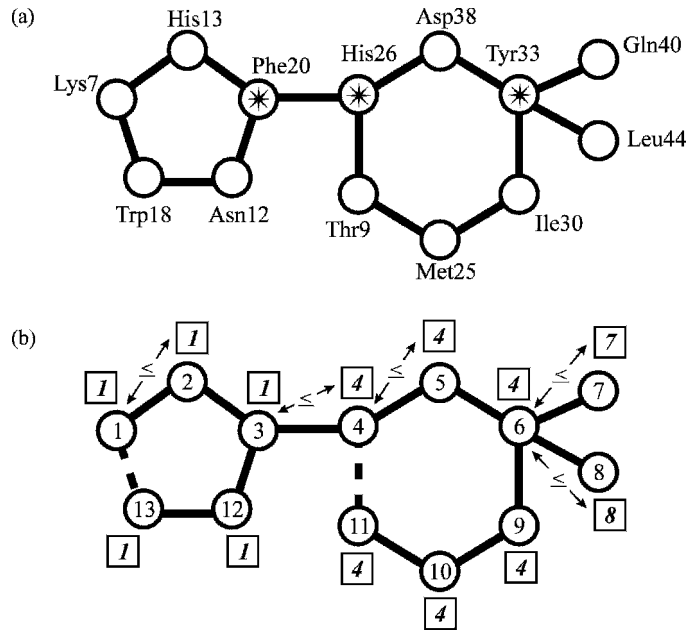


Figure 19 The active side chains are shown in (a). How the side chains are divided into biconnected components is illustrated in (b). Each side chain is numbered with a DFN (circle) and a low number (square next to corresponding residue). A difference between the DFN and the low number indicates the existence of an articulation point for the group of active sidechains. This image was adapted from Canutescu et al.[182]

does not necessarily mean that a residue connected to other residues in a cluster is an articulation point; it is possible for the residues of a cluster to form ring-like structures. The DFN uses the amino acid residues and their connection information to construct a tree (a graph, also known as a residue map) of the amino acid residues in each cluster. The tree is then divided into biconnected components using the DFN numbers and low numbers.[182] The "low number" is assigned to each residue and corresponds to the lowest DFN number that can be reached from a specific amino acid residue based on a path constructed of descendants and a maximum of one back edge (proposed connections between residues that were not explored during the depth-first search). The residues are first explored going down the tree until an endpoint is reached. The search then goes up the tree visiting the residues connected to the already investigated articulation points. The goal of step 6 is to locate a set of active amino acid residue clusters for the final rotamer search.

The seventh and penultimate step used by SCWRL determines which rotamer is most probable for each of the active residues and, accordingly, which should be used in the construction of the GMEC. The best conformation is determined by using a branch-and-bound backtracking method,[182] based on energetics. It provides a quick and reliable way to determine the most probable conformations of residue sidechains for a biopolymer with many different possibilities. The branch-and-bound method makes decisions based on the energetics of residues above and below the energy of the rotamer being considered.[182] Backtracking further optimizes the search by evaluating the number of possible rotamers that exist and the energy of each rotameric state for each residue. For each group of amino acid residues, the residue(s) with the fewest number of rotamers are ranked highest (placed at the root of the tree). Residues containing the greatest number of rotamers are ranked lowest (placed at the bottom of the tree). The concept of ranking the residues based on the number of rotamers and then building a tree to organize them is illustrated in Figure 20. When different residues possess the same number of rotamers, they are ranked from low to high based on their self-energy.[182]

The search for each cluster's GMEC begins by dividing the cluster of amino acid residues into biconnected components that are connected by articulation points (illustrated in Figure 21). The clusters with one articulation point are explored initially. Our discussion here is for clusters with only one articulation point, but the concept can be expanded to accommodate those with two or more articulation points. Each plausible rotameric state for the articulation point is generated, as are the corresponding rotamer states of the remaining residues in the cluster, thus providing multiple conformations for the same cluster. This allows for the construction of multiple low-energy conformations (number of possible rotameric states of the articulation point) of the same cluster based on the different rotameric configurations of the

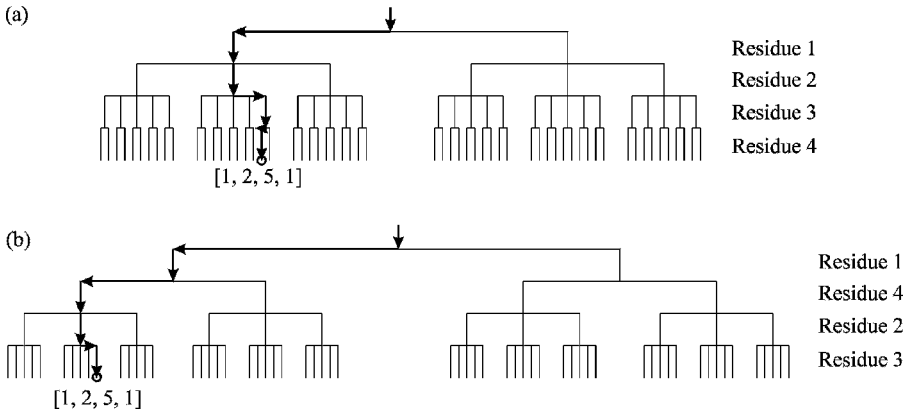
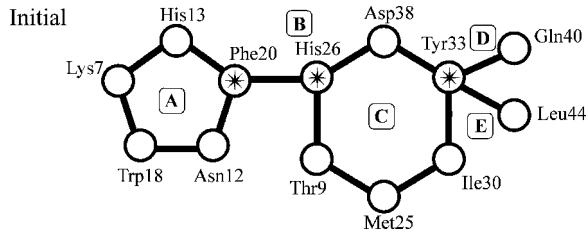
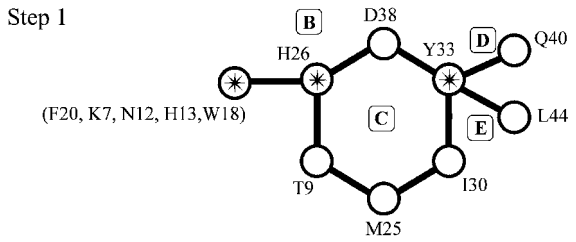


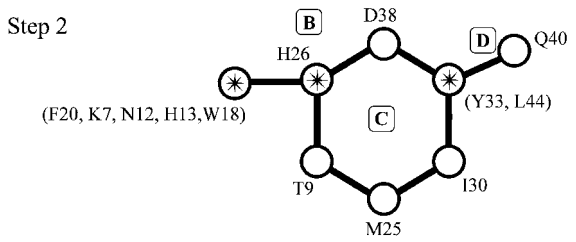
Figure 20 The backtracking trees presented here represent the interaction of four amino acid residues. In (a) the top of the tree is the first residue side chain that possess two possible rotameric states, thus two branches. Residue 2 is on the next level and has three rotamers. The third residue has five possible rotamers, and Residue 4 has two possible rotamer states. The tree in (a) is not efficient because of the number of rotamer options for Residue 3. The same set of residues are evaluated in (b), but the order in which they are examined is changed. Residue 1 is still first, but that is now followed by Residue 4, then Residue 2, and finally Residue 3. Residues having the fewest possible rotamers are evaluated first, thus increasing the speed of the search for the GMEC. A total of 60 possible side-chain interactions exist in this example with the most favorable denoted with a circle [1, 2, 5, 1]. The arrows denote the path of the most favorable rotamer combination. This image was adapted from Canutescu et al.[182]

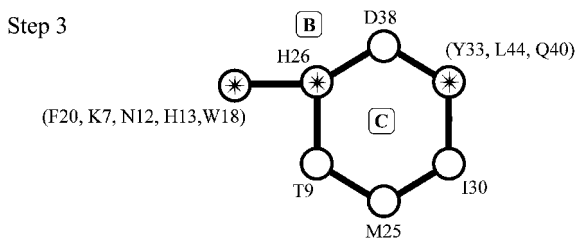
articulation point. The low-energy conformation of all possible rotamers is found using the branch-and-bound backtracking method. The most energetically favorable conformers of the cluster (based on the rotamer conformations of the articulation point) are stored, and the articulation point is converted into a "superrotamer"[182] containing rotamer energies for the residues

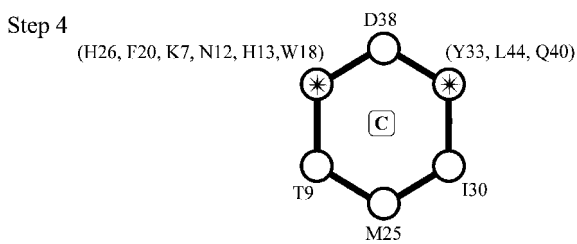
Figure 21 Stepwise solution of a cluster of residues via biconnected components. The GMEC of these side chains is determined through the solution of the individual clusters (biconnected components). The active side chain residues (nodes) are represented as circles, and the side chain's interaction with another side chain(s) is represented as a solid line (edges). As each cluster is solved, it is represented as a superresidue; the residues comprising it are contained in parentheses. The stepwise solving of the cluster starts with the initial set of active side chains that have been divided into biconnected components (denoted as letters within boxes). The collapse of component A creates a superresidue[195] (Step 1) and is followed by the solving of components E (Step 2), D (Step 3). Determination of the GMEC is complete when component B is solved (Step 4). Articulation points are residues that connect groups of active side chains together and are denoted with *. Amino acid residues Phe20 and His26 are first-order articulation points, and Tyr33 is a second-order articulation point. This image was adapted from Canutescu et al.[182]


Initial
His13
B
Asp38
Lys7
Phe20
His26
Tyr33
D
Gln40
A
C
E
Leu44
Trp18
Asn12
Thr9
Ile30
Met25


Step 1
B
D38
H26
Y33
D
Q40
(F20, K7, N12, H13,W18)
C
E
L44
T9
I30
M25


Step 2
B
D38
H26
D
Q40
(F20, K7, N12, H13,W18)
(Y33, L44)
C
T9
I30
M25


Step 3
B
D38
H26
(Y33, L44, Q40)
(F20, K7, N12, H13,W18)
C
T9
I30
M25


Step 4
D38
(H26, F20, K7, N12, H13,W18)
(Y33, L44, Q40)
C
T9
I30
M25

comprising the cluster. Each superrotamer can be thought of as a library of possible side chain orientations based on the possible conformers of the articulation point. Thus, if the articulation point has three possible conformations, only three low-energy conformations of the cluster are constructed and stored in the superrotamer. At this point, one might ask: "If the lowest energy conformation of the cluster is found, what happens when that rotamer is not the best choice for the adjoining set of residues?" To avoid that problem, it is not until the very last cluster is solved that the most appropriate set of rotamers is selected. This methodology does not require every possible conformation of the active rotamers be explored; instead, only combinations of the low-energy states for each cluster are constructed and the GMEC is selected from among these predetermined combinations of conformational states for further analysis.

The final step done by SCWRL is to output the protein model with the most probable side-chain conformations. Included in the output are the residues that were considered to be disulfide bridges, the composition of the rotamer clusters, and a list of the fixed side chains. The use of graph theory in SCWRL is a novel method of solving a combinatorial conformational problem through the segmentation of the amino acid residues into clusters. By identifying the residues that join (articulation point) groups of residues interacting with each other, a divide-and-conquer strategy can be used to solve the GMEC problem. Residues that do not need to partake in the GMEC search are held fixed (i.e., they are omitted from the search to reduce the number of possible conformers). Solving the interaction of amino acid residues in the unique clusters using this reduced set of conformational states enhances computing times significantly.

Energy Minimization

MM is a quick method of removing repulsive contacts between side chains by allowing the side chains to relax to low-energy geometries. Energy minimization of a protein model can be especially useful if the model (target) is closely related to the template structure from which it is cast. Protein models destined for molecular docking simulations or binding site comparisons are often energy minimized to "relax" the amino acid residue side chains (and possibly backbone) before such investigations. Energy minimization is done with the backbone of the protein fixed in space to prevent it from "losing" its secondary structure. The energy minimization produces a local minimum energy structure, not necessarily the global minimum. Energy minimization using MM is comparable with MD with the temperature of the system set to zero Kelvin. It is at the discretion of the investigator to carry out energy minimizations for relaxation of the amino acid side chains given the speed and reliability of rotamer searches.

Molecular Dynamics

Any of the popular MD programs such as AMBER,[173,174] CHARMM,[175,176] Tinker,[177] GROMOS,[40] and NAMD[178] can be used for refinement. The material discussed in this section of the chapter is a simple overview of MD; for more in-depth information on MD methodology and analysis, one is directed to the books by Frenkel and Smit,[199] Field,[200] and Allen and Tildesley.[201] Although the information provided here about MD simulations is basic and generic, it is applicable to any of the aforementioned MD packages. An MD simulation involves three distinct parts: (1) warm-up and equilibrium, (2) sampling the trajectory during a "production" run time period, and (3) analysis of the results. The equilibrium period is essential when preparing the protein for the production portion of the MD simulation. Validating and analyzing the results from the MD simulations provides a way of ensuring that the MD simulations are statistically sound; van Gunsteren and Mark[202] provide a complete review of the processes for analyzing and validating a MD simulation.

The refinement of a protein model with molecular dynamics should be carried out in the medium that best mimics the original environment of the protein being modeled. If the protein being studied is found in the cytoplasm of a cell, the protein is then solvated in a box of explicit water molecules, but if the protein is membrane bound, the environment should mimic the cell wall (water–lipid–water). Performing the MD simulation in a vacuum is a computationally inexpensive way to perform MD, but it will not provide the same quality results as will a simulation using explicit solvent. Although an MD simulation *in vacuo* is not recommended, the use of an implicit solvent treatment might be acceptable in some instances. The refinement methodology described here is for a protein solvated in a box of explicit water molecules.

When preparing for the equilibrium phase of the simulation, it is imperative to ensure that the ionizable amino acid residues are in their correct protonation state, the amide and carboxyl ends are capped with suitable termination groups, and the stereochemistry of the residues is correct (PROCHECK[203] can be used to determine this, as will be discussed). The system is placed into a box of equilibrated water molecules that is large enough to accommodate the long-range cutoff values selected by the user. Different types of water molecules have been developed and parameterized, including, among others, SPC/E,[204] TIP3P,[205] TIP4P,[205,206] and TIP4P-Ew.[207] Because the overall charge of a protein is usually not zero, counterions must be added to balance the charge of the system. The locations of the counterions can be determined by constructing a Coulombic potential grid (usually with 1.0 Å grid spacing) around the protein model. The counterions most frequently used are sodium ions (Na^+) and chloride ions (Cl^-). The counterions are placed at the grid points with the greatest electrostatic potential. After the protein has

been constructed, made charge neutral, and placed in a suitable solvent environment, it is advisable to energy minimize the structure with the same molecular force field being used in the MD simulation to eliminate bad contacts. This energy minimization is needed to avoid the possibility of an unstable MD simulation. An effective and efficient method of accomplishing this is to use a dual-step, sequential-minimization process. The dual-step process refers to the minimization of the solvent (waters and counterions) around a fixed protein structure that in turn is followed by the minimization of the entire system. The sequential-minimization combines a steepest descent energy minimizer to rapidly remove bad contacts followed by a conjugate gradient minimizer to find the locally optimized structure. During the minimization of the system (protein model, water molecules, and counterions) in the second portion of the dual-step minimization, the geometry restraints on the protein model, if they exist, should be released in a step-wise fashion to prevent sudden changes in the structure of the protein that might originate from bad interactions. After the structure has been energy minimized, it is advisable to check the structure of the protein for any obvious irregularities.

Because simple energy minimizations correspond to a zero Kelvin temperature, it is necessary to heat the system to the temperature of the protein being simulated (usually 300 K). The heating process, in which kinetic energy is added over several integration timesteps, often requires that a weak restraint be placed on the protein for the first 20 to 40 ps. Using the Langevin temperature equilibration method[208] to increase and hold the temperature of the system at 300 K is advisable rather than using the Berendsen external heat-bath method.[209] Volume parameters are changed during this process to maintain a constant pressure at 300 K with no restraints imposed on the target protein model for the duration of the 100-ps equilibrium period. The length of a typical MD simulation is hundreds of picoseconds, but it is not uncommon to now see nanosecond time lengths because of advances in both hardware and software. It is advisable to constrain covalent bonds between hydrogen and heavy atoms (removing the high-frequency oscillation from the system), which enables one to increase the time step for the numerical integration process from 1 to 2 fs without detriment to the MD trajectory. To be extra cautious (in addition to reducing the restraints on the protein slowly), it is advisable to increase the integration step size from 0.5 fs to 2 fs (in 0.5-fs steps) over the equilibrium time period to prevent drastic energy changes in the system; this precaution is only needed if the system is extremely heterogeneous. A final equilibrium simulation is conducted with the same parameters as for the first 100 ps but with all constraints on the protein model removed.

At this point, it is necessary to analyze the simulations to ensure that the system is truly at equilibrium. Equilibrium in this case is when the energy, temperature, pressure, volume, and density are constant values over a given time period as is the RMSD of the protein backbone compared with the initial

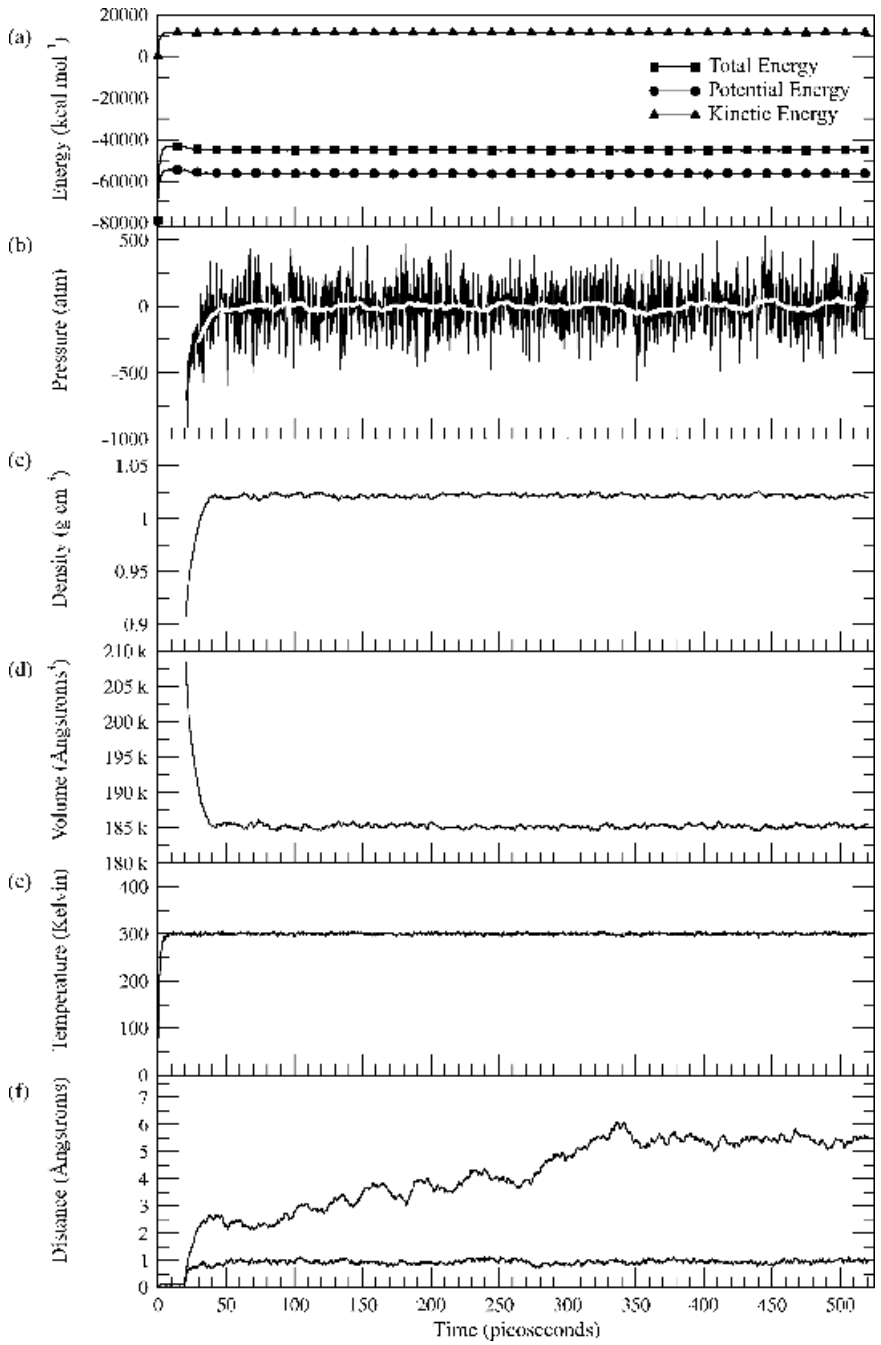
structure. Plotting these parameters (as in Figure 22) versus time allows one to determine visually whether the system is at equilibrium. The potential, kinetic, and total energy terms are interrelated and should be examined. The kinetic energy should maintain a constant value after the heating stage if the thermostat is working correctly and a plot of the "temperature" versus time should show a near-constant value fluctuating about 300 K. It is common to see the potential and total energies rise during the heating phase, and then plateau in their values while the protein's geometric constraints are maintained but then decrease slightly to a constant value after the protein and solvent restraints are removed. After the equilibration, it is expected that the pressure will stabilize to a mean value of 1 atmosphere, but still oscillate about this mean value during the remainder of the simulation. Likewise, as the constraints are removed, plots of the RMSD of protein backbone motion should increase and then oscillate around a constant value; it is acceptable to have the RMSD value increase and then fluctuate between 1 and 2 Å, (as seen in panel F of Figure 22). The parameters being used to gauge whether equilibrium has been reached will eventually arrive at a stable albeit oscillating value, and these constant values are indicative that equilibrium has been achieved and that the production stage of the MD simulation can begin.

MD production runs are just a continuation of the MD equilibrium and continue up to several nanoseconds. The results from the production runs can be used for docking studies or to determine changes in the secondary or tertiary structure of the target protein. Protein models refined with MD simulations consist of a library of structures with similar backbone geometry, but with different side-chain rotamers. These different structures are especially useful for molecular docking studies when the conformation of the protein and the orientation of the side chains are unknown.

Molecular Dynamics with Simulated Annealing

Simulated annealing is an optimization method.[210] It works by heating a system to ensure that many energy states are sampled and then slowly cools the system to ensure that the low-energy structures are found. The system is typically reheated several times, and low-energy conformations are retained for further analysis. Simulated annealing with MD (MD-SA) differs from a simple energy minimization that finds only the minimum that is nearest to its starting point on a potential energy surface.

MD-SA is an effective method for constructing low-energy conformations of the target protein's side chains, as well as for finding alternative secondary and tertiary structures. The simulated annealing protocol removes heat from the system at a slow, controlled rate to ensure that the side chains and other structural features are in energetically favorable positions. Similar to the MD simulation above, there are no hard-and-fast rules for setting up and conducting a MD-SA simulation; the best cooling schedule is tailored for each


(a)
Energy (kcal mol⁻¹)
Total Energy
Potential Energy
Kinetic Energy
(b)
Pressure (atm)
(c)
Density (g cm⁻³)
(d)
Volume (Angstroms³)
(e)
Temperature (Kelvin)
(f)
Distance (Angstroms)
Time (picoseconds)

system. The system is heated to a high temperature, usually between 500 and 800 K. During the heating stage, the backbone structure is restrained to prevent denaturing (unraveling) during the MD simulation. After a short duration at the elevated temperature, the system is slowly cooled before it is completely quenched. The temperature is usually reduced linearly from the initial unrealistic high temperature to approximately 100 K, with increasingly more stringent coupling of the system to the heat bath as the cooling progresses; this stage of the simulation is longest in duration. The final stage is to reduce the temperature of the system to 0 K. This step is done quickly, with tight heat bath coupling and restraints placed on the biopolymer backbone. The final MD steps are similar to a MM energy minimization because the system is approaching 0 K. MD-SA is a quick method to determine several probable side chain conformers and is a good way to sample geometrically feasible shapes of the variable regions of secondary and tertiary structures. The omission of explicit solvent molecules in the MD-SA method is a shortcoming, but it is a necessity because including explicit solvent molecules would be too computationally expensive. Moreover, the kinetic energy pumped into a fully solvated system would be most likely channeled into unproductive solvent rotational/translational motions rather than protein motions that we wish to sample. For these reasons, explicit treatment of solvent is not common in MD-SA refinement of target protein structures.

Both the SCWRL method of determining the most favorable side chain conformation and the MD simulations to explore structure movement are methods that enable the protein model to be refined. The next step in homology modeling is to select the best protein model. When selecting which rendition of a protein model to use, one should not rely on the energy of the refined structure alone because the lowest energy form of a protein structure may not correspond to that of the real system.[211] Selecting the best protein model is aided by the methods discussed in the next section. These methods can also be used to select protein models for further refinement using MD and then reimplemented to ensure that the model is still statistically sound.

Figure 22 The six plots presented here represent the changes in (a) energy, (b) pressure, (c) density, (d) volume, (e) temperature, and (f) RMSD of chicken lysozyme (3lzt[36,97]) in a truncated octahedron of TIP3P water molecules for a duration of 520 ps. The system is considered to be at equilibrium because the kinetic, potential, and total energy of the system (a) remains constant over time. The pressure of the system (b) fluctuates wildly; yet the average remains at approximately 1 atm (white line). The density (c) and volume (d) of the system are not recorded until after 20 ps because of the constraints placed on the system, and they equilibrate at approximately 40 ps. The temperature (e) is constant but fluctuates at 300 K. The RMSD (f) has two different measurements shown. The line fluctuating around 1 Å is the RMSD of the protein structure to illustrate how well its structure is being preserved. The other line illustrates the difference between the initial location of the protein and its location in the truncated octahedron during the simulation. It is expected that the protein will travel during the simulation.

STEP 6: EVALUATING PROTEIN MODELS

Constructing and refining a protein model does not ensure a valid 3-D structure; the construction of the initial 3-D protein model as described in this chapter is a rough process, and the final structures can be structurally (geometrically) improper and energetically flawed.[1] Structurally and energetically flawed models often contain unusual phi and psi (Φ and Ψ) angles (based on Ramachandran plots[212] and statistical data) causing atoms to clash. Even after a successful energy minimization and the reorientation of incorrect dihedral angles, the final minimized structure still might be unsuitable. How can one determine if there are imperfections in the model 3-D protein structure? Several noteworthy methods exist for checking this including PROCHECK,[203,213] Verify3D,[180,181] ERRAT,[214] Protein Volume Evaluation[129] (PROVE), and Protein Structure analysis[179] (ProSa). In addition to statistical and physical property checks provided by these programs, a visual inspection (using programs such as VMD,[215] UCSF Chimera,[99] or the OLDERADO webserver[216]) of superposed protein models determined via NMRCLUST[153] can help locate regions of model variability. We reiterate that although energetic checks can be performed, there is the possibility of the "incorrectly folded" protein structure having a lower potential energy than that of the "correctly folded" protein structure.[211] Many of the methods to be discussed here were developed for the analysis of experimentally determined 3-D protein structures without using energy as a criterion. The primary motive for developing these protein structure analysis methods was to provide scientists with an objective inspection of the quality of experimentally determined protein structures. The protein models constructed in a comparative modeling study can be considered newly solved 3-D protein structures, and they are amendable to analysis by these methods.

PROCHECK

PROCHECK[203,213] performs statistical checks and indicates regions of a protein structure that might require modification because of nonoptimal stereochemistry, which includes planarity, dihedral angles, chirality, nonbonded interactions, mainchain hydrogen bonds, disulfide bonds, and stereochemical assignments. PROCHECK does this using a residue-by-residue analysis. The checking of stereochemical assignments is based on ideal bond lengths and angles contained in the Engh and Huber[217] molecular force field that was constructed from small molecules of the Cambridge Structural Database.[218] PROCHECK provides detailed tables of residue–residue interactions highlighting bad inter-residue contacts and improper bond and torsion angles. The most noticeable output from PROCHECK is its Ramachandran plot[212] (Figure 23). PROCHECK also provides various graphical representations of Ramachandran plots based on residue type. Other plots that can be created

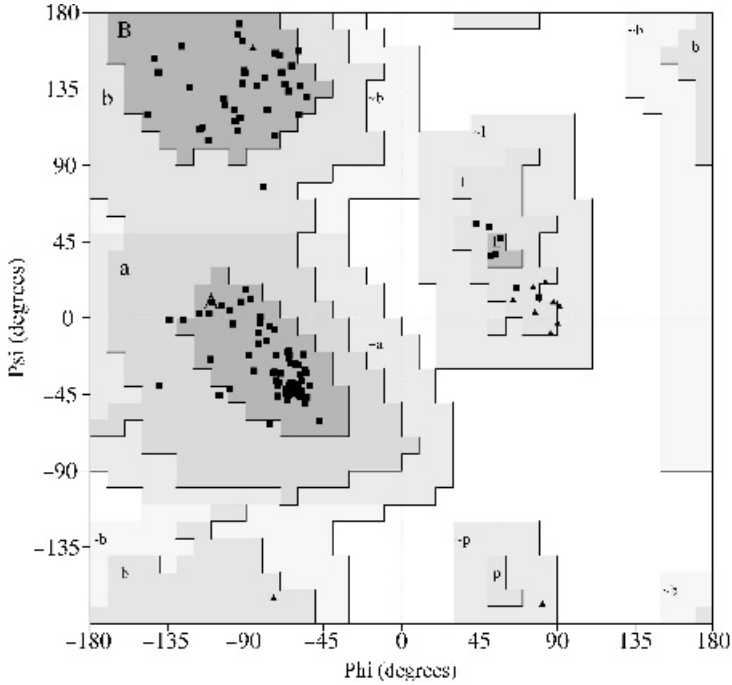


Figure 23 Ramachandran plot for chicken lysozyme created with PROCHECK. The Ramachandran plots are divided into four regions based on the conformation of an amino acid residue's Φ and Ψ angles. The most favored regions (conformations) are labeled with A (core α-helix), B (core β-sheet), and L (core left-handed α-helix). The additionally allowed regions are labeled with a (allowed α-helix), b (allowed β-sheet), l (allowed left-handed α-helix), and p (allowed epsilon α-helix); the generously allowed regions are labeled with ~a (generously allowed α-helix), ~b (generously allowed β-sheet), ~l (generously allowed left-handed α-helix), and ~p (generously allowed epsilon α-helix); disallowed regions are white. Glycine residues are indicated as triangles (▲) because they lack a side chain, which in turn provides them with access to more conformations than other amino acid residues. The overall G-factor score for the chicken lysozyme X-ray structure (3ltz[36,97]) is 0.02, which indicates that the protein is in stereochemical agreement with prescribed values.

include Chi1 versus Chi2 (χ_1 versus χ_2) plots, plots of backbone dihedral angles, side chain dihedral angles, residue properties, backbone bond length distributions, backbone bond angle distributions, RMSD from planarity, and other types of plots. In addition to the statistical stereochemical analysis, PROCHECK includes the ability to rate the quality of the protein structure using geometry factors (G-factors).[219] The G-factor provides a method of determining the "normality" of the stereochemical properties for each residue. The definition of "normal" is based on the analysis of 163 nonhomologous protein structures (sequence homology between any two proteins <35%) solved via X-ray crystallography. The compiled protein structures have a

resolution of 2.0 Å or better, an *R*-factor of 20% or less, and atoms with a zero occupancy value were not included in constructing the definition of "normal" stereochemical properties for residues. The G-factor score comprises log-odd probabilities (therefore unitless) from two classes of protein stereochemistry: torsional angles ($\Phi-\Psi$ and $\chi_1-\chi_2$ distribution, χ_1 only, χ_3 and χ_4 torsion angles, and the Ω torsion angle) and backbone covalent geometries (bond lengths and angles). When analyzing G-factor scores from PROCHECK, protein models with an overall values less than -1.0 are in need of additional analysis to determine the stereochemical deficiencies, whereas protein models with overall G-factor values greater than -0.5 are considered ideal.

Verify3D

Verify3D[180,181] scores the 3-D structure of a protein with probability tables. It assesses the probability that each amino acid residue would occupy in the 3-D structure rather than using the primary structure. Verify3D first converts the tertiary structure into a table, placing the amino acid residues in the rows of the table. The environment of each residue is placed in the first column, and the next 20 columns contain the 3-D to 1-D statistical preferences[181] or probability of each amino acid residue occupying that position in the 3-D structure of the protein. The last two columns of the table consist of the penalty score for opening and extending gaps in the sequence, respectively. The similarity between the most likely 3-D profile and the actual protein structure is plotted (an average of the statistical preferences based on a box size of 21 adjacent residues) versus the sequence number. The higher the overall 3-D to 1-D score, the more sound the model is considered to be.

There are 18 possible environments that an amino acid residue side chain can occupy. The side chains are classified as being buried, partially buried, or solvent exposed based on their solvent-accessible surface areas.[180] The buried and partially buried side chains are further catalogued depending on the environment in which they reside. The buried side chains are divided into three classes, and the partially buried side chains are separated into two classes based on increasing degrees of environmental polarity. The solvent-exposed side chains cannot be subdivided based on environmental polarity because solvent-exposed residues are assumed to experience the same polar environment. The six side chain environments are further classified but not described here. The tables used by Verify3D are similar to mutation tables in PAM,[101–103] BLOSUM,[104] and the Gonnet[111] mutation/similarity matrices and provide a connection between the tertiary structure of a protein and its primary sequence. If a 3-D structure to 1-D sequence average score smaller than zero is obtained for an individual residue (based on a window size of 21 residues), the structure of that residue is considered to be poor. The box size of 21 residues allows for smoothing of local errors and fluctuations.[181] The Verify3D method is highly effective in analyzing comparative protein models because

it compares the model with its amino acid sequence instead of basing the quality of the model on an energy function, such as MM or MD, that might also be used in the construction of the model.[181] There are two acceptable methods for selecting the most likely protein model. The first is to choose the protein model having the greatest average individual amino acid residue score. The other method is to select the protein model with a 3-D to 1-D plot that is similar to the 3-D to 1-D plot of the template protein. Verify3D can also be employed to evaluate the alignment scheme that constructs a protein model. It is done by constructing different models based on varying target-template sequence alignments and then comparing the Verify3D results.[181]

ERRAT

ERRAT evaluates the nonbonded distances of carbon–carbon (CC), carbon–nitrogen (CN), carbon–oxygen (CO), nitrogen–nitrogen (NN), nitrogen–oxygen (NO), and oxygen–oxygen (OO) atoms (sulfur atoms are considered to be oxygen atoms for simplicity). Colovos and Yeates[214] determined that these six distance-related interactions occur in all protein structures, and the authors devised this method originally to assess protein structures derived from ambiguous electron density maps from X-ray studies. ERRAT can locate regions of a protein model that are randomly distributed because of errors in backbone connectivity, alignment, and misplacement of side chains.[214] The average and standard deviation of each type of atom–atom distance, based on known protein structures, is used to evaluate the quality of the protein model.

ERRAT contains a database of acceptable nonbonded atom–atom distances. It classifies the atom–atom distances in a proposed structure and then does a statistical evaluation of those atom–atom distance interactions. The parameters used to construct the database were derived from known protein structures of varying fold classifications. The known structures used for the database were required to have a maximum resolution of 2.5 Å, an *R*-factor less than 25%, be monomeric or homo-oligomeric, be native structures, and contain peptide bond dihedral angles $\pm 15^\circ$ from ideal values based on the secondary structure. A total of 96 solved X-ray protein structures from the PDB[36] were used as the dataset of correct protein structures. The six types of atom–atom distances were restricted to be atoms from different amino acid residues or to be atoms that interact with each other with a through-space distance no greater than 3.5 Å.[214] The average and standard deviation of each atom–atom distance interaction fraction (the fraction of specific atom–atom pairwise distances in a protein structure), as illustrated in Eq. [4][214] for CC distances

$$f(\mathrm{CC}) = \frac{n_{\mathrm{CC}}}{n_{\mathrm{CC}} + n_{\mathrm{CN}} + n_{\mathrm{CO}} + n_{\mathrm{NN}} + n_{\mathrm{NO}} + n_{\mathrm{OO}}} \qquad [4]$$

is calculated and used as a reference to compare nine residue length segments of the protein of interest with values from the database. The distances of the atoms in the four flanking residues on each side of the nine-residue-wide window are measured from the center residue of that window. At least one of the atoms participating in the atom–atom distance calculations must be in the nine-residue window, and only a limited number of distances is calculated for segments that are near structural gaps or loops that exhibit great structural flexibility to remove them from biasing the statistics. One can determine if an amino acid residue (the center residue of the nine-residue window) is in the correct orientation based on the probability of nonbonded interactions experienced by the atoms of that residue. Inclusion of all six atom–atom distances provides the best results. For each amino acid residue, the six nonbonded distances are put into vector form and the probability is calculated as a Gaussian error function. Amino acid residues are considered erroneous when they deviate by 95% or more from their statistically determined nonbonded distances.

The ERRAT method is grounded in the concept that atoms of a protein are not placed randomly throughout the structure. Instead, there are well-defined specific distances between the atoms based on energetic and geometric limitations of the biopolymer. The ERRAT approach is unique compared with the other methods that have been developed to validate whether amino acid residue orientations are correct in a protein that has been built and refined. ERRAT locates residues that do not fit the norm of high-quality crystal structures. Although this method can detect errors as small as 1.5 Å in the backbone configuration, it was originally intended for the analysis of experimentally determined structures that have been correctly refined.[214] It is from this standpoint that ERRAT is best served as a prerefinement method for detecting regions of the protein model containing questionable geometries. It is important to remember that ERRAT identifies flawed amino acid residues; the 95% confidence level is the certainty that a particular residue is imperfect, not that the orientation is correct.

Protein Structure Analysis (ProSa)

The ProSa[88,179,220,221] program uses potentials of mean force (the change in potential energy of a system caused by the variation of a specific coordinate) to locate regions of the protein structure that may contain improper or unsuitable geometries. The 3-D structure of a protein in solution is stabilized through a myriad of forces. The information contained in these forces is extracted from a collection of known structures using Boltzmann's principle[88] and catalogued as a potential of mean force (PMF). A PMF contains all force components typically found in the interaction between atoms plus the effects of the immediate environment (solvent) on the atomic interactions. ProSa derives the PMFs by monitoring the Cα to Cα and the Cβ to Cβ

interactions. This quick method of assessing protein structures allows the user to evaluate the Cα trace (or backbone) of a proposed protein structure, which in turn provides a fast way to appraise the quality of a proposed model before carrying out a time-consuming refinement. In ProSa, the energy of a protein structure is the sum of all pairwise Cα to Cα and Cβ to Cβ residue interactions between amino acid residues at i and j in the protein sequence (e_{ij} corresponds to one measured interaction energy). The sum of all residue pair interactions is the total pair interaction energy (E) obtained by using Eq. [5]:[179]

$$E = \frac{1}{2}\sum_{ij} e_{ij} \qquad [5]$$

The individual energies e_{ij} of the protein depend on its conformation and sequence. The total energy E, therefore, also depends on its conformation and sequence. If the proposed protein model structure is close to the native conformation (and sequence alignment), the energy for that model should be the lowest value when compared with other possible but less viable structures. There are limits on the interaction energy being calculated between residues; the closest interactions cannot be less than 4 Å, and the longest interactions cannot be more than 15 Å away. This interaction shell prevents close-contact interactions from being included in the interaction energy, while excluding long-range interactions that are inherent to larger proteins that might cause imperfections in the interaction energy function. Additionally, the 15 Å cutoff reduces computational costs similar to the method of long-range interaction cutoffs in MD simulations. ProSa uses an energy function derived from PMF information based on pairwise atomic distances gathered from a database of 1353 high-resolution X-ray protein structures.[221] Although the database consists only of soluble globular proteins, ProSa can determine correctly whether proteins from a hydrophobic environment are folded correctly (examples: membrane-bound receptors, integral proteins, and virus coat proteins).[179]

The energies of the protein structure are converted in ProSa into a Z-score (Eq. [6][179]). The best protein model has the lowest Z-score when compared with all other protein structures. The Z-score is composed of three components: the knowledge-based energy of sequence S in a specific conformation C ($E_{S,C}$), the average energy of the sequence in all conformations of a database ($\overline{E}_S$), and the standard deviation (σ_S) associated with the average energy of the sequence:[221]

$$Z_{S,C} = \frac{(E_{S,C} - \overline{E}_S)}{\sigma_S} \qquad [6]$$

Comparing the Z-score of the database with that of a modeled target protein provides a method of determining the viability of the target model. The energy of the modeled target protein is $E_{S,X}$, where X is the observed structure of the

target protein. When the target protein occupies the correct conformation then, $E_{S,X} < E_{S,C}$ and correspondingly $Z_{S,X} < Z_{S,C}$, when the conformation $X \neq C$.[221]

ProSa is not a method for determining close contacts or incorrect stereochemistry. Instead, it is a method for determining whether a 3-D structure is viable. Accordingly, ProSa can quickly screen protein models and determine their merit.[172] In addition, ProSa can be used as a method for assessing the suitability of proposed alignments. Given the speed that protein models can be constructed, ProSa is a viable method for determining which alignment is best. In the current releases of ProSa (ProSa 2003), only the Cβ atomic distance potentials are used because they create better potentials than do the Cα distances.[179,220,221] Wiederstein and Sippl indicate that the specific structural properties of a protein are affected more by the side chains than by the backbone; thus, the use of Cβ atoms produces a more realistic energy potential.[221]

Protein Volume Evaluation (PROVE)

The PROVE[129] method uses the computed volume of individual atoms as a means of evaluating the viability of a protein model. The method was developed initially for checking protein structures that were derived from X-ray crystallography. Because the computed volume is subject to bond lengths, bond angles, and nonbonded interactions, all of those features are optimized during structure refinement. To compute the amino acid residue volume from individual atoms requires knowing the atomic volumes (thus radii) for all atoms in a protein. It is difficult to compile a list of this nature, however, because not all atoms of the same element in a protein are bonded in the same manner; atomic radii depend on the amino acid residue type. The radii of atoms composing cofactors, ligands, and prosthetic groups are difficult to ascertain because of the myriad of possible atom combinations. Even without the different types of non-amino-acid residues to contend with, deriving acceptable atomic volumes can be difficult. A set of 64 high-quality protein structures solved via X-ray crystallography were used by the creators of PROVE to derive atomic volumes. The classic Voronoi method[222] without parameters [as in Surface Volume[223] (SurVol)] was used to calculate the atomic radii. Each atomic volume was calculated based on the polyhedron representing the atom type and bonding configuration. Averaged values were determined for all elements contained in the 64 reference structures. Those atoms were divided into subtypes based on their position in the amino acid residue. The amino acid residue volumes are the summation of the individual atom volumes for each residue. Only buried atoms are computed with PROVE; thus, the derived atomic volumes are based on completely buried residues. Interestingly, only 6% of the residues from the 64 reference structures were buried.[129]

PROVE compares the atomic volumes of a protein model with standard atomic volume values derived from the set of 64 protein structures. Using the same Voronoi methods, the atomic volumes of the protein model under consideration are computed and compared with the "normal" atomic volumes. The difference between the standardized volumes and the protein model being studied is reported as two different types of *Z*-scores. The first *Z*-score involves a comparison between individual atom types as defined in Eq. [7]:[129]

$$Z_i = \frac{\left[V_i^k - \overline{V}^k\right]}{\sigma^k} \qquad [7]$$

In this equation, the volume of the individual atom (i) is V_i^k and k indicates the atom type. The mean volume of that atom type is $\overline{V}^k$, and σ^k is the standard deviation of that atom type.[129] When an atom has a larger than average volume, it has a positive *Z*-score. The opposite is true for a smaller than average atom that would have a negative *Z*-score; the ideal Z-score is zero. The second *Z*-score, called the Z_{RMSD}, involves the comparison of volumes for all atoms or for groups of atoms (in residues). The Z_{RMSD} is calculated as in Eq. [8]:[129]

$$Z_{\text{RMSD}} = \sqrt{\frac{\sum_{i=1}^{N} [Zscore_i]^2}{N}} \qquad [8]$$

where N is the number of atoms and $Z\text{-score}_i$ is the individual atom *Z*-score calculated in Eq. 7. The Z_{RMSD} value is a gauge of deviation for a set of atoms (usually residues) compared with the norm. Again, the *Z*-score and Z_{RMSD} are only calculated for buried protein atoms.

Deviations of atomic volumes do not indicate directly that a defective protein exists because deviations in atomic volumes can be attributed to other physical phenomena. It is for this reason that the authors of PROVE correlated the atom volume deviations with crystallographic qualities of the protein X-ray structure including the resolution (lowest resolvable separation between two carbon atoms), the *R*-factor (measure of how well the refined structure agrees with the experimental model/electron density maps/raw data), and *B*-factors (isotropic temperature factor).[107] A test set of 900 protein structures was constructed, each containing a minimum of 100 buried atoms. The resolution of the protein structures ranged from 1.0 to 3.9 Å. The authors found that for high-resolution structures (1.0 to 1.6 Å), the average Z_{RMSD} was approximately 1.0.[129] When poorer quality crystal structures were considered, the Z_{RMSD} increased. The correlation coefficient for a plot of Z_{RMSD} versus experimental resolution was 0.89 for all protein structures in the test set

and 0.98 for the structures with resolution values between 1.5 and 3.0 Å. The *R*-factor indicates how well the X-ray structure matches experimental scattering data. The correlation between the Z_{RMSD} and the *R*-factor was 76%; this lower value was expected because the *R*-factor is more of an "agreement" factor than it is a gauge of model quality.[129] The correlation between the Z_{RMSD} and the *B*-factor values is 97%. The authors did note that no correlation exists between the *B*-factor and the *Z*-score.

Using PROVE as a protein model evaluator is beneficial because it provides an alternative method of assessing the validity of proposed structures. As demonstrated by the developers of PROVE, it can predict which structures (and regions of structures) are defective based on resolution, *B*-factors, and to a lesser extent *R*-factors, all experimentally derived values that gauge the quality of a structure determined through X-ray crystallography.

Model Clustering Analysis

When building a protein model, it is common to construct many (25 or greater) initial structures and select either the best model or create an averaged structure from that collection (discussed earlier). NMRCLUST,[153] NMRCORE,[224] and OLDERADO[216] are programs that were initially developed for use by scientists solving protein structures by NMR spectroscopy. These programs aid in the superposition and clustering of protein structures. OLDERADO is a combination of the NMRCLUST and NMRCORE methods. The methodologies employed in OLDERADO are discussed below as separate entities devoted to a common task.

The OLDERADO Web server[216] provides a graphical user interface for NMRCLUST and NMRCORE that allows scientists to upload an ensemble of protein structures for analysis. NMRCLUST and NMRCORE examine the ensemble of protein models; first the core atoms are determined, next the domains (regions of similar structure) are defined based on the core atoms, and finally the models are sorted into subfamilies based on conformation. The NMRCORE program examines the variance of the atoms in the ensemble; those with low variance are considered to be in the same domain. The "core" structure of the protein model is not an average of all structures in the ensemble. Instead, the core structure consists of the structural variance of the ensemble. The core is determined from atoms in the backbone that survive a triage based on a dihedral angle order analysis and a penalty function.[216,224] The variances in the pairwise distance of core atoms are then clustered to define domains, which are sections of the protein structure void of flexibility. The traditional definition of a domain is a distinct structural unit of a protein that may have an independent function and may fold into separate compact units. The term domain, as used by OLDERADO, means a rigid section of the protein structure. NMRCLUST performs the pairwise superposition of all protein models in the ensemble. It aligns the structures based on the domains

determined by NMRCORE and then calculates the RMSD of coordinates for each pair. An RMSD matrix is constructed and used in conjunction with a penalty function to cluster the individual models into subfamilies. A major advantage of the OLDERADO suite of applications is that it requires no user intervention.

Aligning protein models based on domains is useful, especially when ensembles of possible target protein structures are collected after model construction or models that have been extracted from MD simulations. Those models containing domains separated by a flexible linker benefit the most from the domain-based pairwise alignment. The superposition of individual domains versus superpositioning the complete protein structure is more intuitive to a scientist, and it provides a realistic representation of the target models' core. To illustrate the importance of the pairwise alignment using domains instead of using all atoms in the models, we consider a protein sampled from a MD trajectory with two distinct domains connected by a flexible segment (linkage). During construction of the protein models, the linkage is treated as a loop. That loop has many different conformations dictating the relationship between the two domains. The angle between the domains can vary from 45° to 180° depending on which loop conformation is selected from the MD ensemble. Constructing an average structure from an ensemble of the sampled structures thus yields a ''blurry'' image of the actual protein that does not accurately portray the ''true'' averaged structure. The resulting structure will instead misrepresent the movement of the two domain regions by using poorly defined structural regions. To avoid this problem, it is better to align domains and use the information from those superpositioned structures to determine the most representative structure of the target protein and then add the flexible connection loop afterward.

OLDERADO clusters the structures of proteins derived from MD simulations into subfamilies based on the conformations of loops and the movement of different protein regions. This clustering is used to select the most representative model from the ensemble of target models constructed. NMRCLUST and NMRCORE can be used individually, but by combining their power and comparing their results with a database of experimentally determined protein family folds, OLDERADO provides additional information about the quality of the final protein model(s). We reiterate that OLDERADO does not construct an average structure, but instead it selects the most representative structure from an ensemble of structures. Clusters of conformationally similar models are created, and the core atoms of protein domains are selected automatically without intervention from the modeler.

The methodologies discussed here differ in how they evaluate both the overall quality and the validity of a protein structure. PROCHECK does this by assessing stereochemistry, Verify3D evaluates the probability of a side chain occupying a specific region, ERRAT assesses the distribution of nonbonded atom–atom interactions for key atoms, ProSa uses PMFs, PROVE

evaluates the quality of a protein structure based on computed volumes, and OLDERADO selects the most representative structure of a collection of structures that have been clustered by superposition of their domains. Presented with these tools for protein model analysis, the immediate question for a novice is which one to use. Regrettably, the best answer to that question is to use them all. One should compare the results from these tools and, where there is disagreement, inspect the data for a consensus at problematic regions and areas where their results differ to make decisions about what to do next. Examining a protein model with programs that rely on the same type of assessment methodology can lead to a false sense of security, so using another method that assesses a model differently is advised. By using several different methods, one can often locate problematic regions that might have otherwise been overlooked. These protein model analysis methods are quick and require little user input; they also possess the ability to run concurrently and have the output condensed into a concise, summary form.

Example: Evaluating Protein Models

The evaluation of protein models is crucial to the selection of the most probable/correct model. Because the programs discussed in this final step use different methodologies, it is not realistic to expect all of them to identify the same set of top models. The results provided in this worked example are for the best and worst BLCA protein models based on the evaluation methods rankings. PROCHECK,[203] Verify3D,[180,181] and ProSa[88,179,220,221] evaluations were performed on the 50 BLCA protein models constructed earlier from the HLCA, CLYC, HBB, and SWM templates. To highlight the dramatic differences between the models created, these models were not refined. One of the most widely known protein evaluation methods is PROCHECK and its widely recognizable Ramachandran plot. In addition to the plot of Φ and Ψ angles is a complete summary of the individual residue's interactions. Ramachandran plots (and the resulting data from a PROCHECK protein evaluation) are a quick way to determine whether the protein model is sound. In Figure 24, we present PROCHECK evaluations of the best and worst models of BLCA that have been constructed from four templates (HLCA, CLYC, HBB, and SWM). The HLCA-based models are presented in (a) of Figure 24. The best model has no residues violating optimal angle conformations, whereas the worst model has only one residue (LYS 16) outside of acceptable values. The best and worst CLYC-based models are shown in (b) of Figure 24; they have three and five residues, respectively, disobeying prescribed stereochemical values. The best and worst HBB-based models in (c) of Figure 24 have 1 and 12 residues outside of the optimal angle values, respectively. The best and worst SWM-based models in (d) of Figure 24 have five and nine residues, respectively, violating good stereochemical angles. The overall G-factor values and the graphical representation of the backbone angles allows the

modeler to quickly find regions of the protein structure that need alignment adjustment or additional refinement to correct poor stereochemical properties. We reiterate that PROCHECK evaluates only the soundness of a protein structure; it does not evaluate residue environment, so PROCHECK cannot distinguish between a correctly and an incorrectly folded protein model.

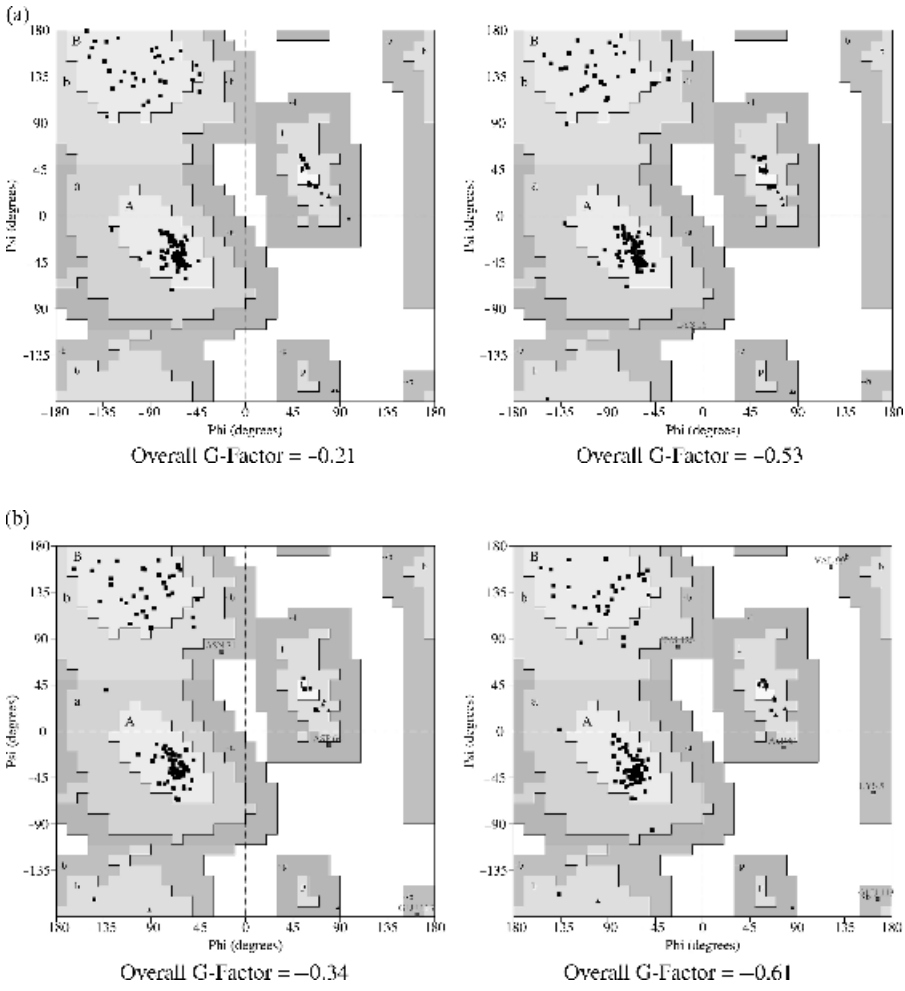


Figure 24 Best and worst bovine α-lactalbumin models based on PROCHECK analysis. Residues that deviate from good stereochemical conformation are denoted with their residue name and number printed above the Φ and Ψ angle marker. The overall G-factor value is an indicator of a protein's stereochemical correctness and is provided for each BLCA model. The best models are on the left, and the worst models are on the right in each example. Panel (a) is human α-lactalbumin-based BLCA; panel (b) is chicken lysozyme-based BLCA; panel (c) is horse hemoglobin β-based BLCA; panel (d) is sperm whale myoglobin-based BLCA.

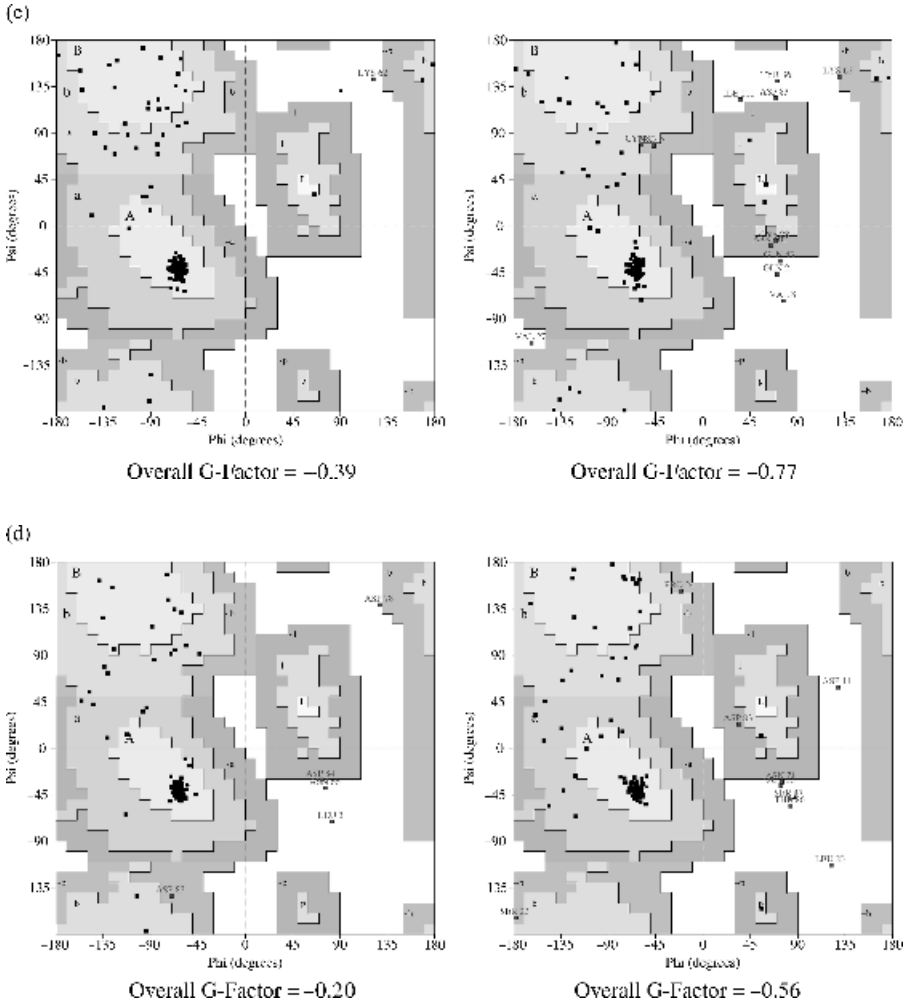


Figure 24 (*Continued*)

The graphical results for the best and the worst BLCA models evaluated with Verify3D[180,181] are provided in Figure 25. Verify3D is a knowledge-based evaluation method that compares the residue's type, secondary structure, and environment with an empirical value. When the Verify3D plots for the best and the worst models are compared with the template structure, regions with significantly different residue composition (based on physicochemical properties) and structure are easily found. It is most notable for the BLCA models created from HBB and SWM, which are all α-helical in nature compared with the α-helical/β-sheet structure of the HLCA and CLYC

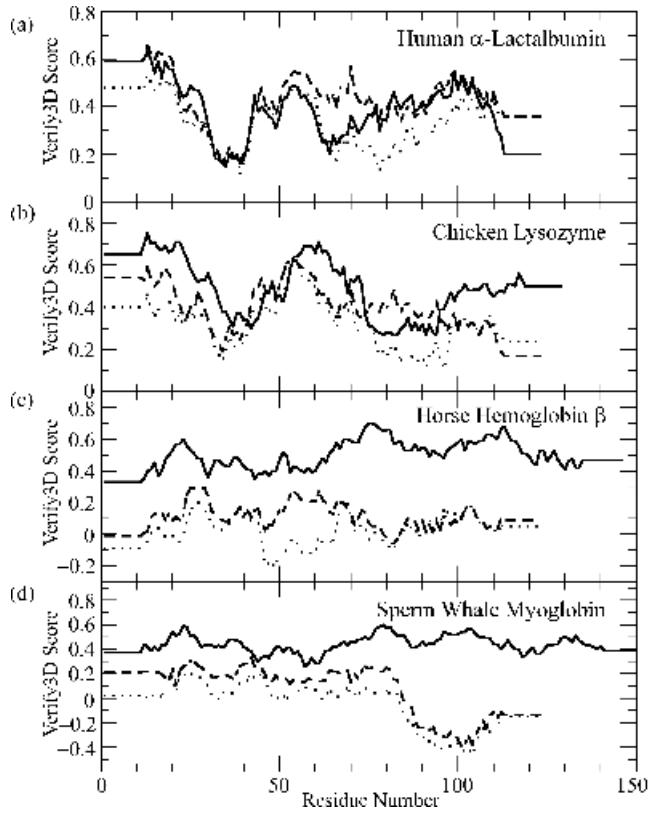


Figure 25 Verify3D plots of X-ray protein structures compared with protein models. The solid lines represent the Verify3D results for the X-ray structure for each protein. The dashed lines are the best, and the dotted lines are the worst BLCA models. A positive Verify3D score for individual residues indicates a favorable residue-structure-environment. The best and worst BLCA models are compared with its template structure (a) HLCA, (b) CLYC, (c) HBB, and (d) SWM.

templates. Additionally, the best and worst BLCA models created from the HLCA and CLYC templates have similar average, minimum, and maximum Verify3D values [(a) and (b) of Figure 25]. Additionally, in (c) and (d) of Figure 25, the Verify3D results of constructing a protein model from dissimilar templates are presented. The BLCA models constructed from HBB and SWM have Verify3D profiles of significantly lower quality than the template structures, and lower overall averages than the BLCA models constructed from HLCA or CLYC (Table 7). The Verify3D profiles possess a similar shape and values to the templates HLCA and CLYC, which indicates that

Table 7 Summary of BLCA Models for Each Template

		PROCHECK		Verify3D				ProSa	
Template	Rank	Model	Overall G-Factor	Model	Min	Ave	Max	Model	Energy
HLCA	1	34	−0.21	42	0.16	0.451	0.66	12	−131.17
	2	28	−0.23	41	0.20	0.438	0.66	32	−129.02
	3	47	−0.25	46	0.12	0.436	0.69	41	−128.88
	4	4	−0.25	17	0.18	0.435	0.63	34	−127.20
	5	27	−0.26	4	0.16	0.423	0.58	40	−127.07
	50	32	−0.53	8	0.12	0.348	0.54	11	−114.58
CLYC	1	34	−0.34	48	0.19	0.460	0.64	9	−116.26
	2	42	−0.35	14	0.23	0.458	0.67	32	−115.18
	3	46	−0.36	36	0.17	0.451	0.66	8	−114.97
	4	40	−0.36	39	0.24	0.447	0.61	25	−113.51
	5	29	−0.36	11	0.24	0.446	0.63	24	−112.23
	50	30	−0.61	31	0.11	0.344	0.62	13	−95.44
HBB	1	42	−0.39	50	−0.01	0.133	0.31	1	9.54
	2	12	−0.42	26	−0.10	0.116	0.31	2	11.11
	3	16	−0.46	20	−0.03	0.113	0.34	20	12.47
	4	18	−0.48	29	−0.06	0.112	0.33	24	13.91
	5	10	−0.48	43	−0.10	0.108	0.35	12	14.92
	50	20	−0.77	33	−0.21	−0.003	0.21	35	50.09
SWM	1	42	−0.20	39	−0.04	0.201	0.34	31	−0.48
	2	31	−0.26	45	−0.02	0.178	0.35	24	2.44
	3	30	−0.31	49	0.04	0.176	0.33	23	3.79
	4	37	−0.32	47	0.02	0.152	0.34	40	4.44
	5	27	−0.32	43	0.01	0.148	0.32	35	4.50
	50	28	−0.56	9	−0.47	−0.036	0.22	5	28.01

good-quality BLCA models were created, whereas BLCA models created from HBB and SWM have poor Verify3D profiles compared with their respective templates.

ProSa[88,179,220,221] is an energy based protein structure evaluation method. The best and the worst BLCA models were evaluated and plotted for comparison with their respective template (Figure 26). The energy of the protein models increases for each of the templates and indicates that the best models were constructed from the HLCA template. One finds little differences between the best and worst models of BLCA and the template of human α-lactalbumin (HLCA) (Figure 26a). A comparison of the BLCA models versus the chicken lysozyme (CLYC) template (Figure 26b) shows a deviation between the models and the template, but the best and worst models are similar. There is a similarity in the overall pattern of the ProSa energy profiles for HLCA and CLYC (models and templates—solid line) and a model of bovine α-lactalbumin (BLCA—dashed line). The BLCA models based on horse

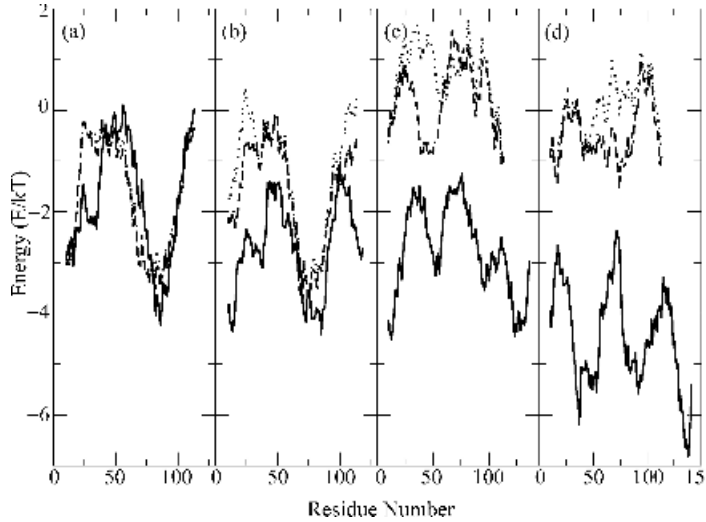


Figure 26 ProSa evaluation of target protein models and their templates. The energy plots are for each of the four BLCA templates and their corresponding best and worst models. The energy profile of the X-ray structure (template) is the solid line, the best BLCA protein model is the dashed line, and the dotted line represents the worst of the 50 BLCA models. Panel (a) is human α-lactalbumin-based BLCA; panel (b) is chicken lysozyme-based BLCA; panel (c) is horse hemoglobin β-based BLCA; panel (d) is sperm whale myoglobin-based BLCA.

hemoglobin β (Figure 26c) and sperm whale myoglobin (Figure 26d) are the worst models whereas the best BLCA models were created from HLCA.

The results of the protein evaluation methods we used to assess the 200 BLCA models in this worked example are summarized in Table 7. The top five and fiftieth BLCA protein models are presented to illustrate the variance between the models created from the same template. A trend that should be noted is the range between the best and worst models for each template. The range between the best and worst BLCA models (based on Verify3D and ProSa evaluations) created with HLCA or CLYC is small compared with models created with HBB or SWM. This small value originates from the alignment of the target (BLCA) to the templates. Better alignments (relating to sequence identity and similarity, and the number and location of gaps) give better models. The small difference is also connected to the similarity of the template's fold to that of the target protein. Here, the use of templates ranging from very similar (73% sequence identity for BLCA to HLCA) to very dissimilar (13% sequence identity for BLCA to SWM), in both sequence identity and 3-D structure (fold), exacerbated the effects of good and poor alignments and overall protein topology. Finally, models created from different templates can be compared directly because they contain the same residues (type and number) in different conformations.

CONCLUSIONS

Comparative modeling uses structural, physicochemical, and functional similarities of proteins to construct the 3-D structure of an unknown protein by using the known structure of another protein(s) as a template. The first step in the construction of a protein model using comparative protein modeling is to find proteins (both sequences and known structures) that are related to the target sequence. The target sequence is then aligned with the related protein sequences (templates included) in a manner to exploit evolutionarily conserved residues and similar structural features. Remember that the alignment of the sequences is a crucial step in comparative modeling, and great care should be taken to include pertinent sequences and templates in the alignment. Many methods exist for aligning sequences. The alignment of the target sequence to the template impacts directly and significantly on the quality and validity of the final protein model. Using a single alignment method is reliable in some cases, but using alignment methods that implement different similarity matrices is better. Once the sequences are aligned and the best template(s) is selected, the target-template alignment will likely need tweaking to move gaps into nonstructural features regions, to ensure that catalytic residues in the target sequence are aligned to their corresponding residues in the template structure, and to preserve structurally conserved regions. With the alignments finalized, the target protein models can then be constructed and refined. Protein model creation depends on the programs available and the number of templates required to obtain the best protein model. The refinement of those models can be as minimal as performing a side chain rotamer search to determine the GMEC, or it can be as involved as carrying out an MD simulation. After refinement, the protein models are evaluated so that the best one(s) can be selected for use in other endeavors. By using several different protein evaluation methods, one can identify a final structure when there is consensus about quality. This ''brute force'' method of examining the protein structures might be viewed as extreme and time consuming, but by using such a vastly different array of assessment methods, the modeler is ensured that a valid, high-quality target protein structure has been created.

The following questions are basic in nature, but they should be addressed by the modeler about the final protein model(s) constructed:

1. What is the sequence identity and sequence similarity between the target and the template?
2. What is the evolutionary distance between the target and the template?
3. If the difference between the target and template exceeds 15%, are there additional sequences bridging the target and template sequences?
4. Are certain residues conserved throughout the sequences and have they been aligned?

5. Do gaps in aligned sequences correspond to regions of random secondary structure?
6. Was the largest variability in sequence identity and sequence similarity between the target and template in the loop regions?
7. Do the secondary structure predictions for the target sequence correspond to the template's secondary structure?
8. If MD was performed on the protein model, are there regions of change in secondary or tertiary structure?
9. Was the largest movement of tertiary structure for the target's structure in loop regions?
10. Can a consensus be inferred about the protein models or specific regions when comparing the results of various evaluation methods?
11. How does the protein model compare with the template protein structure?

Answers to these questions can help you derive a meaningful comparative protein model.

Before one constructs a homology model, it is advantageous to determine what will be the end use of the 3-D model. If the purpose of deriving the protein model is to obtain a general view of the protein's fold, a low-resolution prediction is adequate. If the 3-D structure is to be used in drug design, a structure with little attention given to the loops and side chains could produce a structure that is not only poor, but also misleading.

There are no shortcuts in the comparative modeling process, and attention to detail is paramount. As one becomes more familiar with the field of comparative modeling, one will also become accustomed to specific tools for each step of the target modeling process. Also, as stated, there are many methods for each comparative modeling step. In this chapter, we describe only a few of the mainstream methods and programs. As time progresses, the programs discussed in this chapter may be replaced with others that are faster and more accurate, but the overall process of constructing a protein model via comparative modeling will remain constant.

ACKNOWLEDGEMENT

NIH award P20 GM065805-02 is gratefully acknowledged by JM and DT.

REFERENCES

1. A. R. Leach, *Molecular Modeling: Principles and Applications*, Second Edition, Pearson Education Limited, Harlow, United Kingdom, 2001.
2. C. Branden and J. Tooze, *Introduction to Protein Structure*, Second Edition, Garland Publishing, New York, 1999.

3. W. Browne, A. North, D. Phillips, K. Brew, T. C. Vanaman, and R. L. Hill, *J. Mol. Biol.*, **42**, 65 (1969). A Possible Three-Dimensional Structure of Bovine α-Lactalbumin Based on that of Hen's Egg-White Lysozyme.
4. C. C. F. Blake, G. A. Mair, A. C. T. North, D. C. Phillips, and V. R. Sarma, *Proceedings of the Royal Society of London. Series B, Biological Sciences*, **167**, 365 (1967). On the Conformation of the Hen Egg-White Lysozyme Molecule.
5. M. Levitt, *J. Mol. Biol.*, **226**, 507 (1992). Accurate Modeling of Protein Conformation by Automatic Segment Matching. Available: http://www.bioinformatics.ucla.edu/~genemine/.
6. M. A. Marti-Renom, A. C. Stuart, A. Fiser, R. Sánchez, F. Melo, and A. Šali, *Annu. Rev. Biophys. Biomolec. Struct.*, **29**, 291 (2000). Comparative Protein Structure Modeling of Genes and Genomes. Download MODELLER from http://salilab.org/modeler/.
7. MODELLER: A Program for Protein Structure Modeling, Release 7v7, 2004, Andrej Šali, Department of Biopharmaceutical Sciences, Mission Bay QB3, 1700 4th Street, Suite 503B, University of California-San Francisco, San Francisco, California 94143-2552, email modeler_usage@salilab.org.
8. D. W. Mount, *Bioinformatics: Sequence and Genome Analysis*, Second Edition, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York, 2004. Available: http://www.bioinformaticsonline.org.
9. J. D. Tisdall, *Beginning Perl for Bioinformatics*, O'Reilly, Cambridge, Massachusetts, 2001.
10. J. D. Tisdall, *Mastering Perl for Bioinformatics*, O'Reilly, Cambridge, Massachusetts, 2003.
11. C. Gibas and P. Jambeck, *Developing Bioinformatics Computer Skills*, O'Reilly, Cambridge, Massachusetts, 2001.
12. T. K. Attwood and D. J. Parry-Smith, *Introduction to Bioinformatics*, Prentice-Hall, New York, 1999.
13. T. Lengauer, R. Mannhold, H. Kubinyi, and H. Timmerman, *Bioinformatics – From Genomes to Drugs Volume I: Basic Technologies*, Wiley-VCH, Weinheim, Germany, 2002.
14. T. Lengauer, R. Mannhold, H. Kubinyi, and H. Timmerman, *Bioinformatics – From Genomes to Drugs Volume II: Applications*, Wiley-VCH, Weinheim, Germany, 2002.
15. G. A. Petsko and D. Ringe, *Protein Structure and Function*, New Science Press, Ltd., London, 2003.
16. T. E. Creighton, *Proteins*, Second Edition, W. H. Freeman and Company, New York, 1993.
17. D. L. Nelson, M. M. Cox, and A. L. Lehninger, *Principles of Biochemistry*, Fourth Edition, Worth Publishers, New York, 2004.
18. A. D. Baxevanis, D. B. Davison, R. D. M. Page, G. A. Petsko, L. D. Stein, and G. D. Stormo, *Current Protocols in Bioinformatics*, Vol. 1, John Wiley & Sons, Inc., New York, 2003.
19. A. D. Baxevanis, D. B. Davison, R. D. M. Page, G. A. Petsko, L. D. Stein, and G. D. Stormo, *Current Protocols In Bioinformatics*, Vol. 2, John Wiley & Sons, Inc., New York, 2005.
20. C. Hardin, T. V. Pogorelov, and Z. Luthey-Schulten, *Curr. Opin. Struct. Biol.*, **12**, 176 (2001). *Ab Initio* Protein Structure Prediction.
21. R. Bonneau and D. Baker, *Annu. Rev. Biophys. Biomolec. Struct.*, **30**, 173 (2001). Ab Initio Protein Structure Prediction: Progress and Prospects.
22. J.-E. Shea, M. R. Friedel, and A. Baumketner, in *Reviews in Computational Chemistry*, K. B. Lipkowitz, T. Cundari, and V. Gillet, Eds., Wiley-VCH, New York, Vol. 22, 2006, pp. 169–228. Simulations of Protein Folding.
23. M. Y. Galperin, *Nucleic Acids Res.*, **32**, D3 (2004). The Molecular Biology Database Collection: 2004 update. Available: http://nar.oupjournals.org/cgi/content/abstract/32/suppl_1/D3 Online database at http://www.nar.oupjournals.org/cgi/content/full/32/suppl_1/D3/GKH143TB1.
24. W. Dynan, H. J. Gross, R. I. Gumport, R. B. Hallick, S. M. Linn, A. Maxwell, E. Westhof, J. A. Wise, K. R. Fox, A. R. Kimmel, and A. Bateman, *Nucleic Acids Res.*, **33**, (2005). The 2005 Database Collection. Available: http://www.nar.oupjournals.org.

25. M. Y. Galperin, *Nucleic Acids Res.*, **33**, D5 (2005). The Molecular Biology Database Collection: 2005 update. Available: http://nar.oupjournals.org/cgi/content/abstract/33/suppl_1/D5 Online database at http://nar.oupjournals.org/cgi/content-nw/full/33/suppl_1/D5/TBL1.

26. P. Koehl, in *Reviews in Computational Chemistry*, K. B. Lipkowitz, T. Cundari, and V. Gillet Editors, Wiley-VCH, New York, Vol. 22, 2006, pp. 1–55. Protein Structure Classification.

27. E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R. D. Appel, and A. Bairoch, *Nucleic Acids Res.*, **31**, 3784 (2003). ExPASy: The Proteomics Server for In-depth Protein Knowledge and Analysis. The ExPASy website www.expasy.org.

28. B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider, *Nucleic Acids Res.*, **31**, 365 (2003). The SWISS-PROT Protein Knowledgebase and its Supplement TrEMBL in 2003.

29. A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi, and L.-S. L. Yeh, *Nucleic Acids Res.*, **33**, D154 (2005). The Universal Protein Resource (UniProt). The UniProt website http://www.uniprot.org.

30. N. Hulo, C. J. A. Sigrist, V. L. Saux, P. S. Langendijk-Genevaux, L. Bordoli, A. Gattiker, E. D. Castro, P. Buchner, and A. Bairoch, *Nucleic Acids Res.*, **32**, D134 (2004). Recent Improvements to the PROSITE Database. The PROSITE website http://us.expasy.org/prosite/.

31. *Enzyme Nomenclature: Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzyme-Catalysed Reactions*. International Union of Biochemistry and Molecular Biology (NC-IUBMB). Available: http://www.chem.qmul.ac.uk/iubmb/enzyme/.

32. J. Kopp and T. Schwede, *Nucleic Acids Res.*, **32**, D230 (2004). The SWISS-MODEL Repository of Annotated Three-Dimensional Protein Structure Homology Models. The SWISS-MODEL repository http://swissmodel.expasy.org/repository/.

33. K. Mizuguchi, C. M. Deane, T. L. Blundell, and J. P. Overington, *Protein Sci.*, **7**, 2469 (1998). HOMSTRAD: A Database of Protein Structure Alignments for Homologous Families. The HOMSTRAD database http://www-cryst.bioc.cam.ac.uk/~homstrad/.

34. L. A. Stebbings and K. Mizuguchi, *Nucleic Acids Res.*, **32**, D203 (2004). HOMSTRAD: Recent Developments of the Homologous Protein Structure Alignment Database.

35. A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. L. Sonnhammer, D. J. Studholme, C. Yeats, and S. R. Eddy, *Nucleic Acids Res.*, **32**, D138 (2004). The Pfam Protein Families Database.

36. H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, *Nucleic Acids Res.*, **28**, 235 (2000). The Protein Data Bank. The RCSB/PDB website http://www.rcsb.org/pdb/.

37. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, *J. Mol. Biol.*, **215**, 403 (1990). Basic Local Alignment Search Tool. The NCBI BLAST server http://www.ncbi.nlm.nih.gov/BLAST/.

38. S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, *Nucleic Acids Res.*, **25**, 3389 (1997). Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs.

39. F. Melo and E. Feytmans, *J. Mol. Biol.*, **277**, 1141 (1998). Assessing Protein Structures with a Non-local Atomic Interaction Energy.

40. Biomolecular Simulations: The GROMOS96 Manual and User Guide, 1996, BIOMOS, b.v., Zürich, Switzerland, 1996 Available: http://www.igc.ethz.ch/gromos/.

41. N. Guex and M. C. Peitsch, *Electrophoresis*, **18**, 2714 (1997). SWISS-MODEL and the Swiss-PDB Viewer: An Environment for Comparative Protein Modeling. Download the Swiss-PDB Viewer from http://swissmodel.expasy.org/spdbv/mainpage.htm.

42. T. Schwede, J. Kopp, N. Guex, and M. C. Peitsch, *Nucleic Acids Res.*, **31**, 3381 (2003). SWISS-MODEL: An Automated Protein Homology-modeling Server. The SWISS-MODEL website http://swissmodel.expasy.org/.

43. C. Chothia, A. M. Lesk, M. Levitt, A. G. Amit, R. A. Mariuzza, S. E. V. Phillips, and R. J. Poljak, *Science*, **233**, 755 (1986). The Predicted Structure of Immunoglobulin D1.3 and its Comparison with the Crystal Structure.

44. T. L. Blundell, B. L. Sibanda, M. J. E. Sternberg, and J. M. Thornton, *Nature*, **326**, 347 (1987). Knowledge-Based Prediction of Protein Structures and the Design of Novel Molecules.

45. T. J. Hubbard and T. L. Blundell, *Protein Engineer.*, **1**, 159 (1987). Comparison of Solvent-inaccessible Cores of Homologous Proteins: Definitions Useful for Protein Modeling.

46. J. Greer, *PROTEINS: Structure, Function, and Genetics*, **7**, 317 (1990). Comparative Modeling Methods: Application to the Family of the Mammalian Serine Proteases.

47. M. S. Johnson, N. Srinivasan, R. Sowdhamini, and T. L. Blundell, *Crit. Rev. Biochem. Molec. Biol.*, **29**, 1 (1994). Knowledge-based Protein Modeling.

48. P. A. Bates and M. J. E. Sternberg, *PROTEINS: Structure, Function, and Genetics Supplement*, **37**, 47 (1999). Model Building by Comparison at CASP3: Using Expert Knowledge and Computer Automation.

49. *National Center for Biotechnology Information*. National Library of Medicine & National Institutes of Health, 2005. Available: http://www.ncbi.nlm.nih.gov/.

50. *BLAST Server*. National Center for Biotechnology Information, National Library of Medicine & National Institutes of Health, 2005.

51. A. Bairoch, B. Boeckmann, S. Ferro, and E. Gasteiger, *Briefings in Bioinformatics*, **5**, 39 (2004). Swiss-Prot: Juggling between Evolution and Stability. The Swiss-Prot website http://us.expasy.org/sprot/.

52. C. H. Wu, L.-S. L. Yeh, H. Huang, L. Arminski, J. Castro-Alvear, Y. Chen, Z.-Z. Hu, R. S. Ledley, P. Kourtesis, B. E. Suzek, C. R. Vinayaka, J. Zhang, and W. C. Barker, *Nucleic Acids Res.*, **31**, 345 (2003). The Protein Information Resource. Available: http://pir.georgetown.edu/home.shtml.

53. *Protein Research Foundation Database*. Protein Research Foundation, 2005. Available: http://www4.prf.or.jp/en/index.html.

54. *GenBank*. National Center for Biotechnology Information, National Library of Medicine & National Institute of Health, 2005. Available: http://www.ncbi.nlm.nih.gov/Genbank/index.html.

55. *BLAST Program Selection Guide*. BLAST-Help Group, NCBI User Service, 2005. Available: http://www.ncbi.nlm.nih.gov/BLAST/producttable.shtml.

56. J. Schultz, F. Milpetz, P. Bork, and C. P. Ponting, *Proceedings of the National Academy of Science USA*, **95**, 5857 (1998). SMART, A Simple Modular Architecture Research Tool: Identification of Signaling Domains. Available: http://smart.embl-heidelberg.de/.

57. I. Letunic, R. R. Copley, S. Schmidt, F. D. Ciccarelli, T. Doerks, J. Schultz, C. P. Ponting, and P. Bork, *Nucleic Acids Res.*, **32**, D142 (2004). SMART 4.0: Towards Genomic Data Integration.

58. *Protein Families Database of Alignments and HMMs*. Sanger Institute, 2005. The Pfam website http://www.sanger.ac.uk/Software/Pfam/.

59. R. L. Tatusov, E. V. Koonin, and D. J. Lipman, *Science*, **278**, 631 (1997). A Genomic Perspective on Protein Families. The NCBI COG website http://www.ncbi.nlm.nih.gov/COG/.

60. R. L. Tatusov, N. D. Fedorova, J. D. Jackson, A. R. Jacobs, B. Kiryutin, E. V. Koonin, D. M. Krylov, R. Mazumder, S. L. Mekhedov, A. N. Nikolskaya, B. S. Rao, S. Smirnov, A. V. Sverdlov, S. Vasudevan, Y. I. Wolf, J. J. Yin, and D. A. Natale, *BioMed Central Bioinformat.*, **4**, 41 (2003). The COG Database: An Updated Version Includes Eukaryotes.

61. W. Gish, personal communication, St. Louis, Missouri, 1996–2004. The WU-BLAST website http://blast.wustl.edu.

62. P. E. Bourne, H. M. Berman, B. McMahon, K. D. Watenpaugh, J. Westbrook, and P. M. D. Fitzgerald, in *Meth. Enzymol.*, C. W. Carter, Jr. and R. M. Sweet, Eds., Academic Press, San

Diego, California, Vol. 277, 1997, pp. 571–590. The Macromolecular Crystallographic Information File (mmCIF). Available: http://ndbserver.rutgers.edu/mmcif/.

63. J. Westbrook and P. E. Bourne, *Bioinformatics*, **16**, 159 (2000). STAR/mmCIF: An Extensive Ontology for Macromolecular Structure and Beyond.
64. K. Karplus, R. Karchin, C. Barrett, S. Tu, M. Cline, M. Diekhans, L. Grate, J. Casper, and R. Hughey, *PROTEINS: Structure, Function, and Genetics Supplement*, **45**, 86 (2001). What Is the Value Added by Human Intervention in Protein Structure Prediction? The SAM-T02 website is http://www.cse.ucsc.edu/research/compbio/HMM-apps/T02-query.html.
65. K. Karplus, R. Karchin, J. Draper, J. Casper, Y. Mandel-Gutfreund, M. Diekhans, and R. Hughey, *PROTEINS: Structure, Function, and Genetics*, **53**, 491 (2003). Combining Local-Structure, Fold-Recognition, and New Fold Methods for Protein Structure Prediction.
66. R. Karchin, M. Cline, Y. Mandel-Gutfreund, and K. Karplus, *PROTEINS: Structure, Function, and Genetics*, **51**, 504 (2003). Hidden Markov Models that Use Predicted Local Structure for Fold Recognition: Alphabets of Backbone Geometry. STR information http://www.cse.ucsc.edu/research/compbio/SAM_T02/sam-t02-faq.html.
67. T. Jones and S. Thirup, *EMBO J.*, **5**, 819 (1986). Using Known Substructures in Protein Model Building and Crystallography.
68. A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler, *J. Mol. Biol.*, **235**, 1501 (1994). Hidden Markov Models in Computational Biology: Applications to Protein Modeling.
69. S. R. Eddy, *Curr. Opin. Structural Biol.*, **6**, 361 (1996). Hidden Markov Models.
70. S. R. Eddy, *Bioinformatics*, **14**, 755 (1998). Profile Hidden Markov Models.
71. A. Krogh, B. Larsson, G. von Heijne, and E. L. L. Sonnhammer, *J. Molec. Biol.*, **305**, 567 (2001). Predicting Transmembrane Protein Topology with a Hidden Markov Model: Application to Complete Genomes. Available: http://www.cbs.dtu.dk/services/TMHMM-2.0/.
72. W. Kabsch and C. Sander, *Biopolymers*, **22**, 2577 (1983). Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. Available: http://www.cmbi.kun.nl/gv/dssp/.
73. D. Frishman and P. Argos, *PROTEINS: Structure, Function, and Genetics*, **23**, 566 (1995). Knowledge-Based Secondary Structure Assignment. Available: http://wolf.bi.umist.ac.uk/unix/stride.html.
74. M. Heinig and D. Frishman, *Nucleic Acids Res.*, **32**, W500 (2004). STRIDE: A Web Server for Secondary Structure Assignment from Known Atomic Coordinates of Proteins. Available http://webclu.bio.wzw.tum.de/stride/.
75. *Critical Assessment of Techniques for Protein Structure Prediction*, Protein Structure Prediction Center, Biosciences Directorate, Lawrence Livermore National Laboratory, Livermore, California, Critical Assessment of Techniques for Protein Structure Prediction 6, 2005 Available: http://predictioncenter.llnl.gov/.
76. E. L. L. Sonnhammer, G. von Heijne, and A. Krogh. *Proceedings of the Sixth International Conference on Intellegent Systems for Molecular Biology*, AAAI Press, Menlo Park, California (1998). A Hidden Markov Model for Predicting Transmembrane Helices in Protein Sequences.
77. S. Möller, M. D. R. Croning, and R. Apweiler, *Bioinformatics*, **17**, 646 (2001). Evaluation of Methods for the Prediction of Membrane Spanning Regions.
78. J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, Michigan, 1975.
79. R. Rosenblatt, *Psychological Rev.*, **65**, 386 (1959). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.
80. R. Karchin, *Hidden Markov Models and Protein Sequence Analysis*. University of California at Santa Cruz Bioinformatics (Computational Biology), 2002. Available: http://www.cse.ucsc.edu/research/compbio/ismb99.handouts/KK185FP.html.

81. J. Hargbo and A. Elofsson, *PROTEINS: Structure, Function, and Genetics*, **36**, 68 (1999). Hidden Markov Models That Use Predicted Secondary Structures for Fold Recognition.

82. D. T. Jones, W. R. Taylor, and J. M. Thornton, *Nature*, **358**, 86 (1992). A New Approach to Protein Fold Recognition. Available: http://bioinf.cs.ucl.ac.uk/threader/threader.html.

83. D. T. Jones, R. T. Miller, and J. M. Thornton, *PROTEINS: Structure, Function, and Genetics*, **23**, 387 (1995). Successful Protein Fold Recognition by Optimal Sequence Threading Validated by Rigourous Blind Testing.

84. D. T. Jones, in *Computational Methods in Molecular Biology*, Vol. 32, S. Salzberg, D. Searls, and S. Kasif, Eds., Elsevier Science, New York, 1998, pp. 285–311. THREADER: Protein Sequence Threading by Double Dynamic Programming.

85. D. Tobi and R. Elber, *PROTEINS: Structure, Function, and Genetics*, **41**, 40 (2000). Distance-Dependent, Pair Potential for Protein Folding: Results From Linear Optimization. Available: http://cbsu.tc.cornell.edu/software/loopp/index.htm and http://ser-loopp.tc.cornell.edu/cbsu/loopp.htm.

86. J. Meller and R. Elber, *PROTEINS: Structure, Function, and Genetics*, **45**, 241 (2001). Linear Programming Optimization and a Double Statistical Filter for Protein Threading Protocols.

87. O. Teodorescu, T. Galor, J. Pillardy, and R. Elber, *PROTEINS: Structure, Function, and Bioinformatics*, **54**, 41 (2004). Enriching the Sequence Substitution Matrix by Structural Information.

88. M. J. Sippl, *J. Mol. Biol.*, **213**, 859 (1990). Calculation of Conformational Ensembles from Potentials of Mean Force. An Approach to the Knowledge-Based Prediction of Local Structures in Globular Proteins.

89. A. Godzik, A. Kolinski, and J. Skolnick, *Protein Sci.*, **4**, 2107 (1995). Are Proteins Ideal Mixtures of Amino Acids? Analysis of Energy Parameter Sets.

90. K. Harata, Y. Abe, and M. Muraki, *J. Molec. Biol.*, **287**, 347 (1999). Crystallographic Evaluation of Internal Motion of Human Small, α-Lactalbumin Refined by Full-matrix Least-squares Method.

91. A. C. Pike, K. Brew, and K. R. Acharya, *Structure*, **4**, 691 (1996). Crystal Structures of Guinea-pig, Goat and Bovine α-Lactalbumin Highlight the Enhanced Conformational Flexibility of Regions that are Significant for its Action in Lactose Synthase.

92. T. C. Mueser, P. H. Rogers, and A. Arnone, *Biochemistry*, **39**, 15353 (2000). Interface Sliding As Illustrated by the Multiple Quaternary Structures of Liganded Hemoglobin.

93. D. T. Jones, *J. Molec. Biol.*, **292**, 195 (1999). Protein Secondary Structure Prediction Based on Position-specific Scoring Matrices. Available: http://bioinf.cs.ucl.ac.uk/psipred/.

94. L. J. McGuffin, K. Bryson, and D. T. Jones, *Bioinformatics*, **16**, 404 (2000). The PSIPRED Protein Structure Prediction Server.

95. THREADER 3 User Guide: Protein Fold Recognition by Threading, Version 3 Manual, David T. Jones, Department of Computer Science, Bioinformatics Unit, University College London, Gower Street, London, WC1E 6BT, United Kingdom, email dtj@cs.ucl.ac.uk, 2003.

96. M. A. Walsh, A. McCarthy, P. A. O'Farrell, P. McArdle, P. D. Cunningham, S. G. Mayhew, and T. M. Higgins, *Eur. J. Biochem.*, **258**, 362 (1998). X-ray Crystal Structure of the *Desulfovibrio vulgaris* (Hildenborough) Apoflavodoxin-riboflavin Complex.

97. M. A. Walsh, T. Schneider, L. C. Sieker, Z. Daunter, V. Lamzin, and K. S. Wilson, *Acta Crystallographica, Section D, Biological Crystallography*, **54**, 522 (1998). Refinement of Triclinic Hen Egg-white Lysozyme at Atomic Resolution.

98. J. Thompson, D. Higgins, and T. Gibson, *Nucleic Acids Res.*, **22**, 4673 (1994). CLUSTAL W: Improving the Sensitivity of Progressive Multiple Sequence Alignment Through Sequence Weighting, Position-specific Gap Penalties and Weight Matrix Choice. Clustal W and Clustal X website: http://bess.u-strasbg.fr/BioInfo/ClustalX/Top.html.

99. E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin, *J. Comput. Chem.*, **25**, 1605 (2004). UCSF Chimera – A Visualization System for Exploratory Research and Analysis. Available: http://www.cgl.ucsf.edu/chimera.

100. J. Vojtechovský, K. Chu, J. Berendzen, R. M. Sweet, and I. Schlichting, *Biophysical J.*, **77**, 2153 (1999). Crystal Structures of Myoglobin-Ligand Complexes at Near-Atomic Resolution.

101. M. O. Dayhoff and R. V. Eck, in *Atlas of Protein Sequence and Structure*, Vol. 3, M. O. Dayhoff, Ed., National Biomedical Research Foundation, Washington, D.C., 1968, pp. 33–41. A Model of Evolutionary Change in Proteins.

102. M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt, in *Atlas of Protein Sequence and Structure*, Vol. 5, Suppl. 3, M. O. Dayhoff, Ed., National Biomedical Research Foundation, Washington, D.C., 1978, pp. 345–358. A Model of Evolutionary Change in Proteins.

103. M. O. Dayhoff, W. C. Barker, and L. T. Hunt, in *Methods in Enzymology*, Vol. 91, C. H. W. Hirs and S. N. Timasheff, Eds., Academic Press, New York, 1983, pp. 524–545. Establishing Homologies in Protein Sequences.

104. S. Henikoff and J. G. Henikoff, *Proceedings of the National Academy of Science USA*, **89**, 10915 (1992). Amino Acid Substitution Matrices from Protein Blocks.

105. J. Stoye, *Gene*, **211**, GC45 (1998). Multiple Sequence Alignment with the Divide-and-Conquer Method.

106. S2C: A database correlating sequence and atomic coordinate numbering in the Protein Data Bank, Guoli Wang, Jonathan W. Arthur, and Roland L. Dunbrack, Jr., 2002. Available: http://dunbrack.fccc.edu/Guoli/s2c/.

107. G. H. Stout and L. H. Jensen, *X-Ray Structure Determination: A Practical Guide*, Second Edition, John Wiley & Sons, Inc., New York, 1989.

108. M. A. S. Saqi, R. B. Russell, and M. J. E. Sternberg, *Protein Engineer.*, **11**, 627 (1998). Misleading Local Sequence Alignments: Implications for Comparative Protein Modeling.

109. M. Bajaj and T. L. Blundell, *Annu. Rev. Biophys. Bioengineer.*, **13**, 453 (1984). Evolution and the Tertiary Structure of Proteins.

110. A. J. Jennings, C. M. Edge, and M. J. E. Sternberg, *Protein Engineering Design and Selection*, **14**, 227 (2001). An Approach to Improving Multiple Alignments of Protein Sequences Using Predicted Secondary Structure.

111. G. H. Gonnet, *Handbook of Algorithms and Data Structures: In Pascal and C*, Second Edition, Addison-Wesley Publishing Company, Wokingham, United Kingdom, 1991.

112. G. H. Gonnet, M. A. Cohen, and S. A. Benner, *Science*, **256**, 1443 (1992). Exhaustive Matching of the Entire Protein Sequence Database.

113. S. Henikoff and J. G. Henikoff, *PROTEINS: Structure, Function, and Genetics*, **17**, 49 (1993). Performance Evaluation of Amino Acid Substitution Matrices.

114. S. B. Needleman and C. D. Wunsch, *J. Mol. Biol.*, **48**, 443 (1970). A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins.

115. J. Thompson, T. Gibson, F. Plewniak, F. Jeanmougin, and D. Higgins, *Nucleic Acids Res.*, **25**, 4876 (1997). The CLUSTAL_X Windows Interface: Flexible Strategies for Multiple Sequence Alignment Aided by Quality Analysis Tools.

116. F. Jeanmougin, J. D. Thompson, M. Gouy, D. G. Higgins, and T. J. Gibson, *Trends Biochem. Sci.*, **23**, 403 (1998). Multiple Sequence Alignment with Clustal X.

117. PHYLIP (Phylogeny Inference Package), 3.6, Distributed by Joseph Felenstein. Department of Genome Sciences, University of Washington, Seattle, 2004. Available: http://evolution.genetics.washington.edu/phylip.html.

118. C. Notredame, D. G. Higgins, and J. Heringa, *J. Mol. Biol.*, **302**, 205 (2000). T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment. Available: http://www.ch.embnet.org/software/TCoffee.html.

119. X. Huang and W. Miller, *Adv. Appl. Math.*, **12**, 337 (1991). A Time-Efficient, Linear-Space Local Similarity Algorithm. Available: http://fasta.bioch.virginia.edu/fasta_www/lalign.htm.

120. W. R. Pearson and D. J. Lipman, *Proceedings of the National Academy of Science USA*, **85**, 2444 (1988). Improved Tools for Biological Sequence Comparison. Available: http://www.ebi.ac.uk/fasta33/.

121. W. R. Pearson, in *Methods in Enzymology*, Vol. 183, R. F. Doolittle, Ed., Academic Press, San Diego, California, 1990, pp. 63–98. Rapid and Sensitive Sequence Comparison with FASTP and FASTA.

122. U. Tönges, S. W. Perrey, J. Stoye, and A. W. M. Dress, *Gene*, **172**, GC33 (1996). A General Method for Fast Multiple Sequence Alignment.

123. J. Stoye, *Divide-and-Conquer Multiple Sequence Alignment*, Ph.D. dissertation, Universität Bielefeld, Bielefeld, Germany, 1997.

124. D. J. Lipman, S. F. Altschul, and J. D. Kececioglu, *Proceedings of the National Academy of Science USA*, **86**, 4412 (1989). A Tool for Multiple Sequence Alignment.

125. S. K. Gupta, J. D. Kececioglu, and A. A. Schäffer, *J. Comput. Biol.*, **2**, 459 (1995). Improving the Practical Space and Time Efficiency of the Shortest-Paths Approach to Sum-of-Pairs Multiple Sequence Alignments.

126. M. J. Zvelebil, G. J. Barton, W. R. Taylor, and M. J. E. Sternberg, *J. Molec. Biol.*, **195**, 957 (1987). Prediction of Protein Secondary Structure and Active Sites Using the Alignment of Homologous Sequences.

127. M. Clamp, J. Cuff, S. M. Searle, and G. J. Barton, *Bioinformatics*, **20**, 426 (2004). The Jalview Java Alignment Editor. Available: http://www.jalview.org/.

128. G. J. Barton, *Protein Engineering*, **6**, 37 (1993). ALSCRIPT - A Tool to Format Multiple Sequence Alignments. Request ALSCRIPT from http://www.compbio.dundee.ac.uk/Software/Alscript/alscript.html.

129. J. Pontius, J. Richelle, and S. J. Wodak, *J. Mol. Biol.*, **264**, 121 (1996). Deviations from Standard Atomic Volumes as a Quality Measure for Protein Crystal Structures. Request PROVE from http://www.ucmb.ulb.ac.be/SCMBB/PROVE/.

130. K. B. Nicholas, H. B. Nicholas, Jr., and D. W. Deerfield, II, *EMBNEW.NEWS*, **4**, 14 (1997). GeneDoc: Analysis and Visualization of Genetic Variation. Available: http://www.psc.edu/biomed/dissem/genedoc/index.html.

131. B. Rost, *J. Struct. Biol.*, **134**, 204 (2001). Review: Protein Secondary Structure Prediction Continues to Rise.

132. G. D. Rose, *Nature*, **272**, 586 (1978). Prediction of Chain Turns in Globular Proteins on a Hydrophobic Basis.

133. G. D. Rose and S. Roy, *Proceedings of the National Academy of Science USA*, **77**, 4643 (1980). Hydrophobic Basis of Packing in Globular Proteins.

134. C. Chothia, *J. Mol. Biol.*, **105**, 1 (1976). The Nature of the Accessible and Buried Surfaces in Proteins.

135. J. Kyte and R. F. Doolittle, *J. Mol. Biol.*, **157**, 105 (1982). A Simple Method for Displaying the Hydropathic Character of a Protein.

136. B. Rost and C. Sander, *J. Mol. Biol.*, **232**, 584 (1993). Prediction of Protein Secondary Structure at Better than 70% Accuracy.

137. D. Frishman and P. Argos, *PROTEINS: Structure, Function, and Genetics*, **27**, 329 (1997). Seventy-Five Percent Accuracy in Protein Secondary Structure Predicition.

138. D. T. Jones, *J. Mol. Biol.*, **287**, 797 (1999). GenTHREADER: An Efficient and Reliable Protein Fold Recognition Method for Genomic Sequences.

139. D. Frishman and P. Argos, *Protein Engineer.*, **9**, 133 (1996). Incorporation of Long-Distance Interactions into a Secondary Structure Prediction Algorithm.

140. M. Ouali and R. D. King, *Protein Sci.*, **9**, 1162 (2000). Cascaded Multiple Classifiers for Secondary Structure Prediction. Available: http://www.aber.ac.uk/~phiwww/prof/.

141. D. T. Jones, W. R. Taylor, and J. M. Thornton, *Biochemistry*, **33**, 3038 (1994). A Model Recognition Approach to the Prediction of All-Helical Membrane Protein Structure and Topology. Available: http://bioinf.cs.ucl.ac.uk/psipred/.

142. D. T. Jones, *FEBS Lett.*, **423**, 281 (1998). Do Transmembrane Protein Superfolds Exist?

143. B. Rost, in *Methods in Enzymology: Computer Methods for Macromolecular Sequence Analysis*, Vol. 266, R. F. Doolittle, Ed., Academic Press, San Diego, California, 1996, pp. 525–539. PHD: Predicting One-Dimensional Protein Structure by Profile-Based Neural Networks. Available: http://cubic.bioc.columbia.edu/predictprotein/.

144. M. S. Johnson, M. J. Sutcliffe, and T. L. Blundell, *J. Mol. Evolution*, **30**, 43 (1990). Molecular Anatomy: Phyletic Relationships Derived from Three-Dimensional Structures of Proteins.

145. J. Felenstein, *Annu. Rev. Genet.*, **22**, 521 (1988). Phylogenies from Molecular Sequences: Inference and Reliability.

146. M. F. Perutz, *J. Mol. Biol.*, **13**, 646 (1965). Structure and Function of Haemoglobin. I. A Tentative Atomic Model of Horse Oxyhaemoglobin.

147. M. F. Perutz, J. C. Kendrew, and H. C. Watson, *J. Mol. Biol.*, **13**, 669 (1965). Structure and Function of Haemoglobin. II. Some Relations between Polypeptide Chain Configuration and Amino Acid Seqeunce.

148. M. F. Perutz, H. Muirhead, J. M. Cox, and L. C. G. Goaman, *Nature*, **219**, 131 (1968). Three-Dimensional Fourier Synthesis of Horse Oxyhaemoglobin at 2.8 Å Resolution: The Atomic Model.

149. T. F. Havel and M. E. Snow, *J. Mol. Biol.*, **217**, 1 (1991). A New Method for Building Protein Conformations from Sequence Alignments with Homologues of Known Structure.

150. N. Srinivasan and T. L. Blundell, *Protein Engineer.*, **6**, 501 (1993). An Evaluation of the Performance of an Automated Procedure for Comparative Modeling of Protein Tertiary Structure.

151. S. M. Brocklehurst and R. N. Perham, *Protein Sci.*, **2**, 626 (1993). Prediction of the Three-dimensional Structures of the Biotinylated Domain from Yeast Pyruvate Carboxylase and of the Lipoylated H-protein from the Pea Leaf Glycine Cleavage System: A New Automated Method for the Prediction of Protein Tertiary Structure.

152. A. Šali and T. L. Blundell, *J. Mol. Biol.*, **234**, 779 (1993). Comparative Protein Modeling by Satisfaction of Spatial Restraints.

153. L. A. Kelley, S. P. Gardner, and M. J. Sutcliffe, *Protein Engineer.*, **9**, 1063 (1996). An Automated Approach for Clustering an Ensemble of NMR-Derived Protein Structures into Conformationally-Related Subfamilies. Available: http://neon.ce.umist.ac.uk/nmrclust/protocol.html.

154. A. Fiser, R. K. Do, and A. Šali, *Protein Sci.*, **9**, 1753 (2000). Modeling of Loops in Protein Structures.

155. MODELLER: A Program for Protein Structure Modeling, Release 8v0, 2005, Andrej Šali, Department of Biopharmaceutical Sciences, Mission Bay QB3, 1700 4th Street, Suite 503B, University of California-San Francisco, San Francisco, California 94143-2552, email modeler_usage@salilab.org.

156. Molecular Operating Environment, 2004.03, Chemical Computing Group Inc., 1010 Sherbrooke Street West, #910, Montreal, Quebec, Canada H3A 2R7, 2004. Available: http://www.chemcomp.com.

157. P. A. Bates, L. A. Kelley, R. M. MacCallum, and M. J. E. Sternberg, *PROTEINS: Structure, Function, and Genetics Supplement*, **45**, 39 (2001). Enhancement of Protein Modeling by Human Intervention in Applying the Automatic Programs 3D-JIGSAW and 3D-PSSM. The 3D-JIGSAW server http://www.bmm.icnet.uk/servers/3djigsaw/. The 3D-PSSM server http://www.sbg.bio.ic.ac.uk/~3dpssm/index2.html.

158. B. Contreras-Moreira and P. A. Bates, *Bioinformatics*, **18**, 1141 (2002). Domain Fishing: A First Step in Protein Comparative Modeling.

159. L. A. Kelley, R. M. MacCallum, and M. J. E. Sternberg, *J. Mol. Biol.*, **299**, 499 (2000). Enhanced Genome Annotation using Structural Profiles in the Program 3D-PSSM.

160. D. Fischer, C. Barret, K. Bryson, A. Elofsson, A. Godzik, D. Jones, K. J. Karplus, L. A. Kelley, R. M. MacCallum, K. Pawowski, B. Rost, L. Rychlewski, and M. J. E. Sternberg, *PROTEINS: Structure, Function, and Genetics Supplement*, **37**, 209 (1999). CAFASP-1: Critical Assessment of Fully Automated Structure Prediction Methods.

161. L. A. Kelley, R. M. MacCallum, and M. J. E. Sternberg. *Third Annual Conference on Computational Molecular Biology (RECOMB 99)*, 1999,The Association for Computing Machinery, New York. Recognition of Remote Protein Homologies Using Three-Dimensional Information to Generate a Position Specific Scoring Matrix in the Program 3D-PSSM.

162. P. A. Bates, R. M. Jackson, and M. J. E. Sternberg, *PROTEINS: Structure, Function, and Genetics Supplement*, **29**, 59 (1997). Model Building by Comparison: A Combination of Expert Knowledge and Computer Automation.

163. R. Sanchez and A. Šali, *Curr. Opin. Struct. Biol.*, **7**, 206 (1997). Advances in Comparative Protein-Structure Modeling.

164. S. Srinivasan, C. March, and S. Sudarsanam, *Protein Sci.*, **2**, 277 (1993). An Automated Method for Modeling Proteins on Known Templates Using Distance Geometry.

165. A. Šali, J. P. Overington, M. S. Johnson, and T. L. Blundell, *Trends Biochem. Sci.*, **15**, 235 (1990). From Comparisons of Protein Sequences and Structures to Protein Modeling and Design.

166. J. M. Thornton, *J. Mol. Biol.*, **151**, 261 (1981). Disulphide Bridges in Globular Proteins.

167. W. Braun and N. Gō, *J. Mol. Biol.*, **186**, 611 (1985). Calculation of Protein Conformations by Proton-Proton Distance Constraints. A New Efficient Algorithm.

168. A. D. MacKerell, Jr., D. Bashford, M. Bellott, R. L. Dunbrack, Jr., J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, I. W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiórkiewicz-Kuczera, D. Yin, and M. Karplus, *J. Phys. Chem. B*, **102**, 3586 (1998). All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins.

169. M. J. Sutcliffe, I. Haneef, D. Carney, and T. L. Blundell, *Protein Engineer.*, **1**, 377 (1987). Knowledge Based Modeling of Homologous Proteins, Part I: Three-dimensional Frameworks Derived from the Simultaneous Superposition of Multiple Structures.

170. C. M. Topham, A. McLeod, F. Eisenmenger, J. P. Overington, M. S. Johnson, and T. L. Blundell, *J. Mol. Biol.*, **229**, 194 (1993). Fragment Ranking in Modeling of Protein Structure: Conformationally Constrained Environmental Amino Acid Substitution Tables.

171. M. J. Sutcliffe, F. R. Hayes, and T. L. Blundell, *Protein Engineer.*, **1**, 385 (1987). Knowledge Based Modeling of Homologous Proteins, Part II: Rules for the Conformations of Substituted Sidechains.

172. Comparative Modeling Perl Scripts, Troy Wymore, Biomedical Initiative Group, Pittsburgh Supercomputing Center, Carnegie Mellon University, 4400 Fifth Avenue, Pittsburgh, PA 15213, 2005. Available: http://www.psc.edu/biomed/research/biostr/perl.htm.

173. D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham, III, S. DeBolt, D. Ferguson, G. Seibel, and P. A. Kollman, *Comput. Phys. Commun.*, **91**, 1 (1995). AMBER: A Package of Computer Programs for Applying Molecular Mechanics, Normal Mode Analysis, Molecular Dynamics and Free Energy Calculations to Simulate the Structural and Energetic Properties of Molecules.

174. AMBER, Version 8, Department of Pharmaceutical Chemistry, Box 2280, University of California-San Francisco, 600 16th Street, San Francisco, California 94143-2280, 2004. Available: http://amber.scripps.edu.

175. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comput. Chem.*, **4**, 187 (1983). CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations.

176. A. D. MacKerell, Jr., B. R. Brooks, C. L. Brooks, III, L. Nilsson, B. Roux, Y. Won, and M. Karplus, in *Encyclopedia of Computational Chemistry*, Vol 1, P. v. R. Schleyer, Ed., John Wiley & Sons, Inc., Chichester, United Kingdom, 1998, pp. 271–277. CHARMM: The Energy Function and Its Parameterization with an Overview of the Program.

177. Tinker, 4.2, Jay William Ponder, Washington University School of Medicine, St. Louis, Missouri, 2004. Available: http://dasher.wustl.edu/tinker/.

178. L. Kalé, R. Skell, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulten, *J. Comput. Phys.*, **151**, 283 (1999). NAMD2: Greater Scalability for Parallel Molecular Dynamics. Download NAMD from http://www.ks.uiuc.edu/Research/namd/.

179. M. J. Sippl, *PROTEINS: Structure, Function, and Genetics*, **17**, 355 (1993). Recognition of Errors in Three-Dimensional Structures of Proteins. Download ProSa from http://www.came.sbg.ac.at/Services/prosa.html.

180. J. U. Bowie, R. Lüthy, and D. Eisenberg, *Science*, **253**, 164 (1991). A Method to Identify Protein Sequences That Fold into a Known Three-Dimensional Structure. The Verify3D server http://www.doe-mbi.ucla.edu/Services/Verify_3D/.

181. R. Lüthy, J. U. Bowie, and D. Eisenberg, *Nature*, **356**, 83 (1992). Assessment of Protein Models with Three-Dimensional Profiles.

182. A. A. Canutescu, A. A. Shelenkov, and R. L. Dunbrack, Jr., *Protein Sci.*, **12**, 2001 (2003). A Graph-Theory Algorithm for Rapid Protein Side-Chain Prediction. Download SCWRL from http://dunbrack.fccc.edu/SCWRL3.php.

183. J. Janin, S. J. Wodak, M. Levitt, and B. Maigret, *J. Mol. Biol.*, **125**, 357 (1978). Conformation of Amino Acid Side-Chains in Proteins.

184. M. N. G. James and A. R. Sielecki, *J. Mol. Biol.*, **163**, 299 (1983). Structure and Refinement of Penicillopepsin at 1.8Å Resolution.

185. M. J. McGregor, S. A. Islam, and M. J. E. Sternberg, *J. Mol. Biol.*, **198**, 295 (1987). Analysis of the Relationship Between Side-Chain Conformation and Secondary Structure in Globular Proteins.

186. J. W. Ponder and F. M. Richards, *J. Mol. Biol.*, **193**, 775 (1987). Tertiary Templates for Proteins: Use of Packing Criteria in the Enumeration of Allowed Sequences for Different Structural Classes.

187. P. Koehl and M. Delarue, *J. Mol. Biol.*, **239**, 249 (1994). Application of a Self-consistent Mean Field Theory to Predict Protein Side-chains Conformation and Estimate Their Conformational Entropy.

188. S. C. Lovell, J. M. Word, J. S. Richardson, and D. C. Richardson, *PROTEINS: Structure, Function, and Genetics*, **40**, 389 (2000). The Penultimate Rotamer Library. Available: http://kinemage.biochem.duke.edu/databases/rotamer.html.

189. Z. Xiang and B. H. Honig, *J. Mol. Biol.*, **311**, 421 (2001). Extending the Accuracy Limits of Prediction for Sidechain Conformation. Available: http://trantor.bioc.columbia.edu/programs/sidechain/index.html.

190. M. P. Jacobson, R. A. Friesner, Z. Xiang, and B. H. Honig, *J. Mol. Biol.*, **320**, 597 (2002). On the Role of the Crystal Environment in Determinig Protein Side-chain Conformations.

191. R. L. Dunbrack, Jr., *Curr. Opin. Struct. Biol.*, **12**, 431 (2002). Rotamer Libraries in the 21st Century. Available: http://dunbrack.fccc.edu/bbdep/.

192. R. L. Dunbrack, Jr. and M. Karplus, *J. Mol. Biol.*, **230**, 543 (1993). Backbone-Dependent Rotamer Library for Proteins: Application to Side-Chain Predictions.

193. R. L. Dunbrack, Jr. and M. Karplus, *Nature Structural Biol.*, **1**, 334 (1994). Conformational Analysis of the Backbone-Dependent Rotamer Preferences of Protein Sidechains.

194. R. L. Dunbrack, Jr. and F. E. Cohen, *Protein Sci.*, **6**, 1661 (1997). Bayesian Statistical Analysis of Protein Side-Chain Rotamer Preferences.

195. R. F. Goldstein, *Biophys. J.*, **66**, 1335 (1994). Efficient Rotamer Elimination Applied to Protein Side-Chains and Related Spin Glasses.

196. L. L. Looger and H. W. Hellinga, *J. Mol. Biol.*, **307**, 429 (2001). Generalized Dead-End Elimination Algorithms Make Large-Scale Protein Side-Chain Structure Prediction Tractable: Implications for Protein Design and Structural Genomics.

197. J. Desmet, M. De Maeyer, B. Hazes, and I. Lasters, *Nature*, **356**, 539 (1992). The Dead-End Elimination Theorem and Its Use in Protein Side-Chain Positioning.

198. D. B. Gordon and S. L. Mayo, *J. Comput. Chem.*, **19**, 1505 (1998). Radical Performance Enhancements for Combinatorial Optimization Algorithms Based on the Dead-End Elimination Theorem.

199. D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, Second Edition, Academic Press, New York, 2001. Available: http://molsim.chem.uva.nl/frenkel_smit/.

200. M. J. Field, *A Practical Introduction to the Simulation of Molecular Systems*, Cambridge University Press, Cambridge, United Kingdom, 1999.

201. M. P. Allen and D. J. Tildesley, *Computer Simulations of Liquids*, Oxford University Press, New York, 1989.

202. W. F. van Gunsteren and A. E. Mark, *J. Chem. Phys.*, **108**, 6109 (1998). Validation of Molecular Dynamics Simulation.

203. R. A. Laskowski, M. W. MacArthur, D. S. Moss, and J. M. Thornton, *J. Appl. Crystallogr.*, **26**, 283 (1993). PROCHECK: A Program to Check the Stereochemical Quality of Protein Structures. Download PROCHECK from http://www.biochem.ucl.ac.uk/~roman/procheck/procheck.html.

204. H. J. C. Berendsen, J. R. Grigera, and T. P. Straatsma, *J. Phys. Chem.*, **91**, 6269 (1987). The Missing Term in Effective Pair Potentials.

205. W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, and M. L. Klein, *J. Chem. Phys.*, **79**, 926 (1983). Comparison of Simple Potential Functions for Simulating Liquid Water.

206. W. L. Jorgensen and J. D. Madura, *Mol. Phys.*, **56**, 1381 (1985). Temperature and Size Dependences for Monte Carlo Simulations of TIP4P Water.

207. H. W. Horn, W. C. Swope, J. W. Pitera, J. D. Madura, T. J. Dick, G. L. Hura, and T. Head-Gordon, *J. Chem. Phys.*, **120**, 9665 (2004). Development of an Improved Four-Site Water Model for Biomolecular Simulations: TIP4P-Ew.

208. R. W. Pastor, B. R. Brooks, and A. Szabo, *Mol. Phys.*, **65**, 1409 (1988). An Analysis of the Accuracy of Langevin and Molecular Dynamics Algorithms.

209. H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. D. Nola, and J. R. Haak, *J. Chem. Phys.*, **81**, 3684 (1984). Molecular Dynamics with Coupling to an External Bath.

210. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Science*, **220**, 671 (1983). Optimization by Simulated Annealing.

211. M. Levitt, *J. Mol. Biol.*, **104**, 59 (1976). A Simplified Representation of Protein Conformations for Rapid Simulation of Protein Folding.

212. G. N. Ramachandran and V. Sasisekharan, in *Advances in Protein Chemistry*, Vol. 23, C. B. Anfinsen, Jr., M. L. Anson, J. T. Edsall, and F. M. Richards, Eds., Academic Press, New York, 1968, pp. 283–438. Conformation of Polypeptides and Proteins.

213. A. L. Morris, M. W. MacArthur, E. G. Hutchinson, and J. M. Thornton, *PROTEINS: Structure, Function, and Genetics*, **12**, 345 (1992). Stereochemical Quality of Protein Structure Coordinates.

214. C. Colovos and T. O. Yeates, *Protein Sci.*, **2**, 1511 (1993). Verification of Protein Structures: Patterns of Nonbonded Atomic Interactions. Download ERRAT from http://www.doe-mbi.ucla.edu/Services/ERRATv2/ and for more information visit http://www.doe-mbi.ucla.edu/People/Yeates/Gallery/Errat.html.

215. W. Humphrey, A. Dalke, and K. Schulten, *J. Mol. Graph. Model.*, **14**, 33 (1996). VMD–Visual Molecular Dynamics. Download VMD from http://www.ks.uiuc.edu/Research/vmd/.

216. L. A. Kelley and M. J. Sutcliffe, *Protein Sci.*, **6**, 2628 (1997). OLDERADO: On-Line Database of Ensemble Representatives and Domains. The OLDERADO server http://neon.ce.umist.le.ac.uk/olderado/index.html.

217. R. A. Engh and R. Huber, *Acta Crystallographica, Section A*, **A47**, 392 (1991). Accurate Bond and Angle Parameters for X-Ray Protein Structure Refinement.

218. The Cambridge Structural Database, 2003, Cambridge Crystallographic Data Centre, 12 Union Road, Cambridge, United Kingdom, 2003. Available: http://www.ccdc.cam.ac.uk/prods/csd/csd.html.

219. M. W. MacArthur, R. A. Laskowski, and J. M. Thornton, *Curr. Opin. Struct. Biol.*, **4**, 731 (1994). Knowledge-based Validation of Protein Structure Coordinates Derived by X-Ray Crystallography and NMR Spectroscopy.

220. M. J. Sippl, *Curr. Opin. Struct. Biol.*, **5**, 229 (1995). Knowledge-Based Potentials for Proteins.

221. M. Wiederstein and M. J. Sippl, *J. Mol. Biol.*, **345**, 1199 (2005). Protein Sequence Randomization: Efficient Estimation of Protein Stability Using Knowledge-Based Potentials.

222. G. F. Voronoi, *J. für die Reine und Angewandte Mathematik*, **134**, 198 (1908). Nouvelles Applications des Paramètres Continus à la Théorie des Formes Quadratiques.

223. P. Alard, *Calculs de Surface et d'Énergie dans le Domain des Macromolécules*, Service de Conformation des Macromolécules Biologiques et de Bioinformatique, Service de Conformation des Macromolécules Biologiques et de Bioinformatique, B-1050 Bruxelles, Belgium, 1991.

224. L. A. Kelley, S. P. Gardner, and M. J. Sutcliffe, *Protein Engineer.*, **10**, 737 (1997). An Automated Approach for Defining Core Atoms and Domains in an Ensemble of NMR-Derived Protein Structures. Available: http://neon.ce.umist.le.ac.uk/nmrcore/coreprot.html.

CHAPTER 3

Simulations of Protein Folding

Joan-Emma Shea,[a*] Miriam R. Friedel,[ab] and Andrij Baumketner[ac]

[a]*Department of Chemistry and Biochemistry, The University of California, Santa Barbara, CA*
[b]*Department of Physics, The University of California, Santa Barbara, California*
[c]*Institute for Condensed Matter Physics, Lviv, Ukraine*

INTRODUCTION

Proteins are the fundamental building blocks of life. They play an essential role in a wide variety of ways from antibodies fighting infection, to enzymes catalyzing biochemical reactions, to the structural collagen in our bones. Synthesized on the ribosome as linear chain of amino acids, a protein must quickly and spontaneously find a unique three-dimensional structure known as the native state to perform its function. The process by which a protein goes from this unstructured chain to its native state is known as protein folding, and it is arguably one of the most critical processes in biology.

The native state of a single-domain protein is its tertiary structure, and it is estimated that approximately 1000 different shapes or folds exist in nature.[1] The native state generally adopts a globular three-dimensional shape, stabilized by both covalent (peptide bonds, disulfide bridges) and noncovalent (electrostatic, hydrophobic and hydrogen bonds) interactions. Local structures within the native fold are known as secondary structures, with common motifs including alpha helices and beta sheets. The classic experiments of Anfinsen et al.[2–4] suggested a ''Thermodynamic Hypothesis'' for folding, in which the

Reviews in Computational Chemistry, Volume 22
edited by Kenny B. Lipkowitz, Thomas R. Cundari, and Valerie J. Gillet

native state of the protein corresponds to its thermodynamically most stable configuration. Their in vitro studies demonstrated that a completely denatured (unfolded) protein will refold spontaneously to its biologically active three-dimensional conformation if placed in a native-like environment. This finding implies that all information needed by a protein to find its native state is found in its sequence. Even so, the ability of scientists to predict a protein's structure from its amino acid sequence remains elusive.

Although having the ability to predict a protein's structure from its sequence (and subsequently, to design proteins with a specific structure and function) is of primary interest to scientists and engineers, understanding the nature of the folding process is equally important. As alluded to, two requirements must be met for a protein to fold. First, the native state of a protein must be thermodynamically stable at biologically relevant temperatures. Its lowest energy state must be the native one, and if perturbed, the protein must be able to refold easily. Second, the protein must fold on the order of milliseconds to seconds at temperatures where it is thermodynamically stable. Understanding the factors that contribute to folding rates and thermodynamic stability are critical for understanding the protein folding process.

In 1968, Cyrus Levinthal noted that if a protein had to search randomly through all of its possible conformations, its folding time would be on the order of the age of the universe.[5] His argument, which has subsequently become known as Levinthal's paradox, is as follows. Suppose that the conformational space of a protein is composed of four distinct $\phi\Psi$ angles for each peptide bond. For a modest size 100-residue protein, this is 4^{100}, or approximately 10^{60}, possible conformations. If the protein can convert from one conformation to the next in a mere 10^{-12} seconds (a reasonable estimate), it would still take 10^{48} seconds for the protein to fold, longer than the current age of the universe. Thus, Levinthal proposed that each protein must follow a specific pathway to its native state, going through the same sequence of conformational changes in the same order, each and every time it folds.

Levinthal's paradox remained a stumbling block for "protein folders" until the late 1980s, when a new view of protein folding began to emerge called the energy landscape perspective. First quantified by Bryngelson and Wolynes,[6,7] energy landscape theory postulates that not every conformation of a protein is of equal statistical weight and that as a protein folds, it is making its way down a funnel-shaped free energy landscape, with its native state at the bottom of that funnel. As the protein folds, it has a bias to find conformations of lower free energy; thus, it need not take the same path to its native state each time. Energy landscape theory is a statistical description of protein folding and was developed using tools from polymer physics, replica methods, and so on.[8] Furthermore, this theory can explain both kinetic and thermodynamic aspects of protein folding. We will discuss more specifics about energy landscape theory in the following section of this tutorial, but we also refer readers to an excellent review by Dill et al.[9] that more closely examines

differences between energy landscape theory and the ''sequential micropath view'' postulated by Levinthal.

In conjunction with energy landscape theory and independent of it, researchers have proposed models to explain the specifics of the folding process including the diffusion-collision (or framework), hydrophobic collapse, classical nucleation, and nucleation condensation models. One of the earliest of these models, which appeared before the development of energy landscape theory, is the framework (or diffusion collision) model.[10–12] This model postulates that local elements of native secondary structure (such as an alpha helix or beta sheet) would form first, independent of a protein's tertiary structure. These secondary structure elements, also called microdomains, would diffuse until they collided, stuck together, and caused the tertiary structure of the protein to form. The hydrophobic collapse model[9,13,14] postulates that proteins first collapse around their hydrophobic residues and then rearrange to form specific secondary structures. This process is presumably driven by native-like tertiary interactions, which implies that secondary and tertiary structural elements do not necessarily form independent of each other. The classic nucleation model postulates that some secondary structural elements form first, with nucleation seeded by only a few residues. Formation of protein structure subsequently propagates out from this nucleus.[15,16] A related but slightly different protein folding model is the nucleation condensation mechanism.[17] This model postulates the formation of a weak local nucleation site that is stabilized by long-range interactions, which thus effectively creates an extended nucleus. Nucleation and overall structure formation occur concurrently, which leads to a highly cooperative folding process.

Computer simulations are well suited for investigating the diversity of folding scenarios allowed by energy landscape theory, as well as for testing the predictions of the folding models discussed above. Simulations have provided enormous insight into the folding process[18] over the past two decades, and as computers continue to improve, the detail with which we can study the folding process will also improve. We now provide a brief introduction to the different types of computer simulations, providing more detail in the section on protein models.

Three main types of computer simulations exist that are commonly used to study protein folding: lattice models, off-lattice minimalist models, and fully atomic simulations. In lattice models, each amino acid is modeled as a single entity, often called a bead, and those beads constituting the protein are confined to move on a lattice via the Metropolis Monte Carlo method.[19] Because these models are so simplistic, they can be studied in exhaustive detail and have provided insight into the physics behind the protein folding mechanism. Although lattice models have proved useful for studying protein folding, they are inherently unrealistic because beads do not model well the complexity of amino acids and Monte Carlo simulations do not provide realistic kinetic information about folding. Off-lattice models, although still relatively simplistic,

offer a slightly more realistic picture of the folding process. In these models, amino acids can be modeled by a single bead (representing the alpha carbon in the protein backbone) or by multiple beads (in an attempt to represent φ–ψ angles). The proteins in these models are not confined to move on a lattice, but instead they can move freely in space. Their dynamics are derived by integrating Newton's equations of motion to recover more realistic dynamics, and both thermodynamic and kinetic information can be obtained from this method.

Fully atomistic simulations are the most realistic of the three simulation methods. They include a fully detailed description of the amino acids comprising the protein, and they are thus much more true to life than the other models. In addition, solvent molecules may be added explicitly or implicitly to the simulation. Because of this extreme detail, a simulation of a small protein may require the treatment of thousands of atoms. Fully atomic simulations are thus extremely computationally expensive, and only short time scales can be explored. As computational power continues to increase, so do the time scales accessible with this method. Nevertheless, fully atomic simulations still cannot capture kinetic information; they are, however, useful in understanding important local interactions that drive protein folding.

We have gained a tremendous amount of knowledge about protein folding through a combination of theoretical, computational, and experimental approaches throughout the past four decades. Still, many aspects of this process remain that need to be explored further. In this chapter, we will focus primarily on the theoretical and computational techniques used to study the protein folding problem, as well as the results from these studies. We will begin with a discussion of the theoretical framework of protein folding, focusing on energy landscape theory, and then placing a particular emphasis on the thermodynamics and kinetics of folding. From there, we will continue with a discussion of simulation models and techniques, as well as more advanced folding topics that are used to describe the nature of the transition state ensemble for folding. Finally, we will comment on our outlook on the field and promising future directions in our concluding remarks.

THEORETICAL FRAMEWORK

Energy Landscape Theory

Much of our modern understanding of protein folding is based on the study of the underlying "energy landscape" for folding. This energy landscape framework offers an alternative view to the older sequential folding framework and accounts for both the thermodynamic and the kinetic requirements for folding. Its statistical mechanical treatment emphasizes the many configurations available to the protein and allows for the existence of multiple pathways for folding.

The theoretical basis for the energy landscape framework stems from work on spin glass systems. These magnetic systems, studied extensively by Derrida et al., consist of randomly arranged spins that can interact both ferromagnetically and anti-ferromagnetically.[20–22] Because of these competing interactions, not all spin orientations can be satisfied mutually, resulting in what is known as "frustration."[23] A consequence of this frustration is an underlying "rough" energy landscape characterized by deep energy minima separated by high barriers. This type of landscape shows a characteristic phase transition at the glass transition temperature T_g, the temperature at which the system finds itself trapped in a given low-energy state.[24] A similar energy landscape exists for random heteropolymers (RHPs), which are polymers whose sequence consists of a random letter code of amino acids. A one-dimensional rough energy landscape is represented in Figure 1a. RHP systems do not have a well-defined ground state, and their many low-energy (but structurally

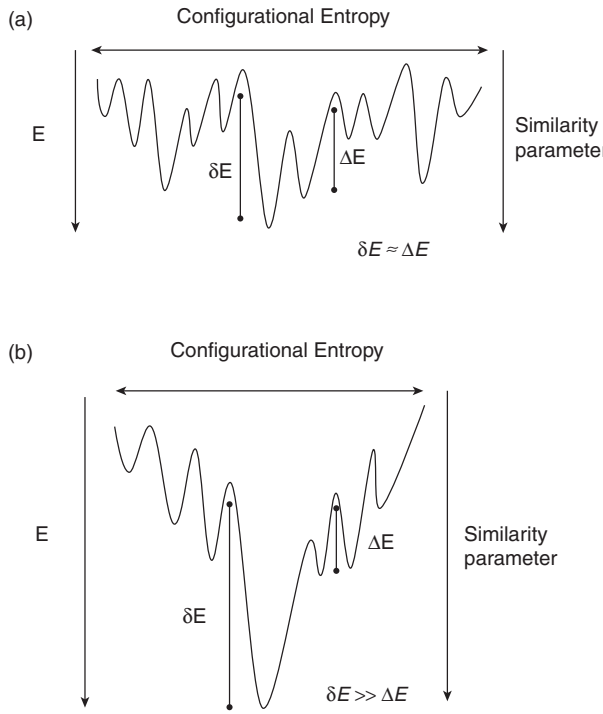


Figure 1 One-dimensional representation of (a) a rough energy landscape that is typical of frustrated sequences. The landscape is characterized by numerous low-energy minima separated by high-energy barriers. The energetic bias δE to the global minimum is of the same magnitude as the roughness of the surface ΔE ($\delta E \sim \Delta E$). (b) A funneled energy landscape of a foldable sequence. Here $\delta E \gg \Delta E$ and the native state is both thermodynamically and kinetically accessible.

dissimilar) conformations can interconvert only by overcoming large energy barriers. There are two sources of frustration in RHP systems. The first source is "energetic," which results from unfavorable interactions between different amino acids caused, for instance, by a nonspecific collapse of the chain induced by the hydrophobic effect. The second source of frustration is "topological," originating from the constraints associated with polymeric chain connectivity.[25] Approximating the energy states as random independent energies, Bryngelson and Wolynes applied the random energy model (REM), first introduced by Derrida et al. in the context of spin glasses,[26,27] to describe the energy landscape of RHP.[6,7] The REM is exactly solvable for RHP when energy levels are both random and uncorrelated. An alternative, yet equivalent, statistical mechanical treatment for RHP is based on mean-field replica methods. It was developed independently by Garel and Orland[28] and Shakhnovich and Gutin[29] at the same time as the development of the REM method. RHPs, like their spin glass analogs, undergo a phase transition as the temperature is lowered. This "freezing" transition occurs at a temperature T_g, where the system runs out of entropy (the so-called "entropy crisis") and becomes trapped in one of the low-energy states.

Proteins differ from RHP by possessing a unique stable ground-state conformation, accessible on biological time scales. Rough energy surfaces are not compatible with protein folding, because such surfaces do not satisfy either kinetic or thermodynamic criteria for folding. Consider the energy landscape in Figure 1a. A slight perturbation (temperature or pH change, for instance) can cause the lowest energy state to rise in energy, leading to a new, structurally different state with lower energy. Proteins are only biologically active in a given state, in which, for example, the binding site is capable of interacting with a ligand. This conformation must be immutable to environmental fluctuations and correspond to the global energy minimum. The rough energy landscape does not satisfy these requirements. From a kinetic standpoint, the rough surface also falls short of rationalizing protein folding: Folding times on such surfaces are governed by the rate of escape from the deep energy traps, a timescale comparable with the random search time of Levinthal's paradox.

Protein sequences have evolved to minimize unfavorable interactions leading to frustrated, rough surfaces.[8] A viable energy landscape that accounts for kinetic and thermodynamic aspects of folding can be achieved through a "funneled" surface, with a global energy minimum corresponding to the native state that is well separated from the other local energy minima on the surface.[30] An energy surface is characterized by three elements: the ruggedness of the surface given by the fluctuations in energy ΔE, the stability of the ground state δE, and the configurational entropy S_o. An energy surface is multidimensional, but it is represented for simplicity in one-dimension in Figure 1b. Folding is initiated at the top of the funnel, with the protein existing in an unfolded state. The configurational entropy (the width of the funnel) is

large, reflecting the numerous possible unfolded conformations available to the protein. As folding proceeds, the protein descends the funnel, getting trapped transiently in the many local free energy minima (on the order of k_BT) riddling the surface. These minima correspond to the latent frustration of the surface. Because a part of the protein's frustration is "topological" and hence cannot be removed, even an ideally designed sequence with no conflicting energetic interactions will retain some frustration.[31–33] The large energy gap δE separating collapsed states from the lowest energy folded state provides a driving force (energetic bias) toward that native state. The funneled landscape allows for a multiplicity of folding routes, and it accounts for both the thermodynamic (single ground state) and the kinetic requirements (folding on biological timescales) for folding. Analytical studies were performed by Wolynes et al. who treated the simplest energy landscape for a protein. They assumed (1) that the unfavorable (non-native) contacts possess random energy contributions and can hence be treated as RHP using REM and (2) that, on average, the total energy of the protein decreases as favorable (native) contacts are formed.[6] These assumptions lead to an overall bias toward the native state and form the basis for the "principle of minimum frustration" for proteins. Two main transition temperatures are associated with folding on a funneled surface: the folding temperature T_f at which the global energy minimum (the folded state) becomes stable and the glass transition temperature T_g. Analytical studies have demonstrated that well-designed protein sequences possess a large energy bias δE compared with the roughness of the surface ΔE, or said equivalently, they have a large T_f/T_g ratio.[6] The rough energy landscape in Figure 1a is an extreme case in which $\delta E \sim \Delta E$, which corresponds to a nonfoldable scenario.

Thermodynamics and Kinetics of Folding: Two-State and Multistate Folders

Energy landscapes provide a microscopic description for folding and are best probed by single molecule experiments, in which the folding of a single protein chain is monitored from the unfolded state to the folded state.[34–36] Such experiments are still in their infancy, however, and most protein folding experiments to date are performed in the "bulk" and provide averaged macroscopic, rather than microscopic, information on folding. Bulk experiments consider thermodynamic (macro) states, such as the native, intermediate, and unfolded states, which consist of a collection or "ensemble" of single-chain conformations (microstates).[37,38] A macroscopic description of folding involves projecting the total free energy F(Q) of the protein system as a function of one (or more) reaction coordinate Q, as opposed to the microscopic description in which only the free energy of a single-chain E(Q) is considered. The macroscopic and microscopic energies are related by $F(Q) = E(Q) - TS_o(Q)$, where S_o is the configurational entropy.[37] We note

that the solvent-averaged microscopic energy E is in reality a free energy because it contains entropic contributions from the solvent molecules.

The thermodynamics and kinetics of folding obtained from bulk experiments can be singularly simple despite the structural complexity of proteins and their large number of degrees of freedom. The averaged data obtained from these experiments can be reconciled in terms of reaction models, similar to those used in chemical kinetics of small molecules.[39] These models range from two-state models in which only the unfolded and folded states are populated, to more complex multistate models in which one or more intermediate states are present. A two-state model seems to be the rule rather than the exception for describing the folding of single-domain proteins with fewer than 100 amino acids. For larger proteins, multistate models are used because they show multistate folding behavior.[40]

The free energy of a two-state folder as a function of a reaction coordinate for different temperatures is given schematically in Figure 2. Folding is cooperative, and the free energy is characterized by two minima, corresponding to the unfolded and folded states, separated by a small barrier originating from the incomplete cancellation of the competing (favorable) energetic and (unfavorable) entropic contributions to folding. The two minima are equal at the folding temperature T_f, with the folded and unfolded states populated to the same extent. Structures residing at the top of the barrier correspond to the transition state ensemble. At temperatures below T_f, the native state becomes more populated, whereas at temperatures above T_f, the unfolded state dominates. A caveat of this type of reaction model is that ''the'' reaction coordinate for folding is assumed to be known. In particular, transition state structures can only be identified from the free energy profiles if such a reaction coordinate is known and identifying this reaction coordinate (if it even exists) is problematic because of the thousands of degrees of freedom associated with the protein system. The nature of the transition state ensemble and the shortcomings of reaction-based models of folding will be discussed in the section ''Advanced Topics: The Transition State Ensemble for Folding.''

The thermodynamic and kinetic differences between two- and multistate folders are illustrated in Figure 3. For simplicity, a three-state model with a single intermediate (I) is considered in this example. Experimentally, two-state thermodynamics can be inferred from both spectroscopic data and calorimetric melting curves.[41] A signature of ''all-or-none'' transitions between the unfolded and the folded states is the equivalence between calorimetrically determined enthalpies ΔH_{cal} and van't Hoff derived enthalpies ΔH_{vh}: $\Delta H_{cal} = \Delta H_{vh}$. Processes involving intermediates show $\Delta H_{cal} > \Delta H_{vh}$. A second thermodynamic signature is the existence of identical transition curves (plots of unfolded population as a function of temperature) that are obtained by using different spectroscopic probes. For instance, far ultra violet circular dichroism (UVCD) spectra (a measure of secondary structure) should match near-UVCD spectra (a probe of tertiary structure). Near- and far-UVCD spectra

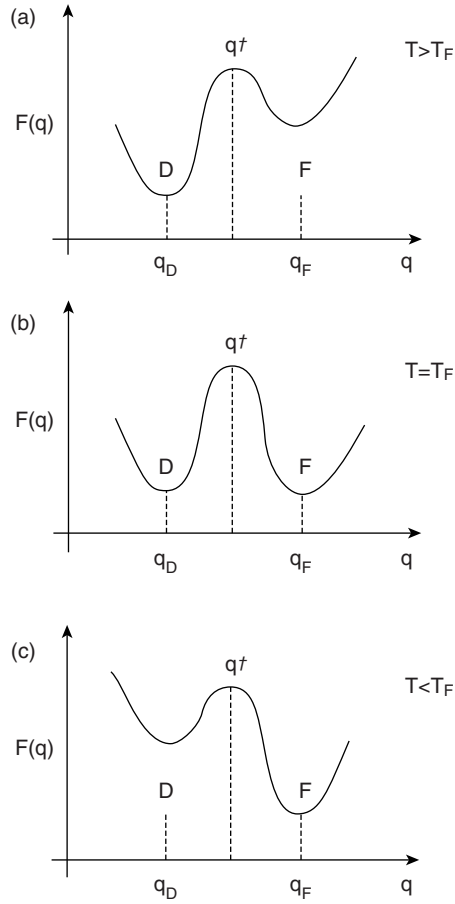


Figure 2 Free energy surfaces for a two-state folder as a function of a reaction coordinate q, which is plotted at different temperatures. The folded (F) and denatured (D) states are separated by a free energy barrier located at $q^{\dagger}$. (a) Above the folding transition temperature ($T > T_f$), the entropic contribution of the free energy dominates and the unfolded state is favored. (b) At the folding transition temperature ($T = T_f$), both the unfolded and the folded states are equally populated. (c) Below the transition temperature ($T < T_f$), the favorable energetic contributions to folding dominate and the native basin is favored.

that do not change in concert indicate the presence of an intermediate state. From a kinetic standpoint, two-state folders display single exponential kinetics, consistent with a rate process involving the crossing of a single dominant barrier. Multistate folders, on the other hand, possess free energy profiles with additional minima corresponding to the intermediate states and thus have nonexponential folding kinetics. The kinetics of folding is typically probed experimentally using a stopped-flow apparatus. One first unfolds the protein

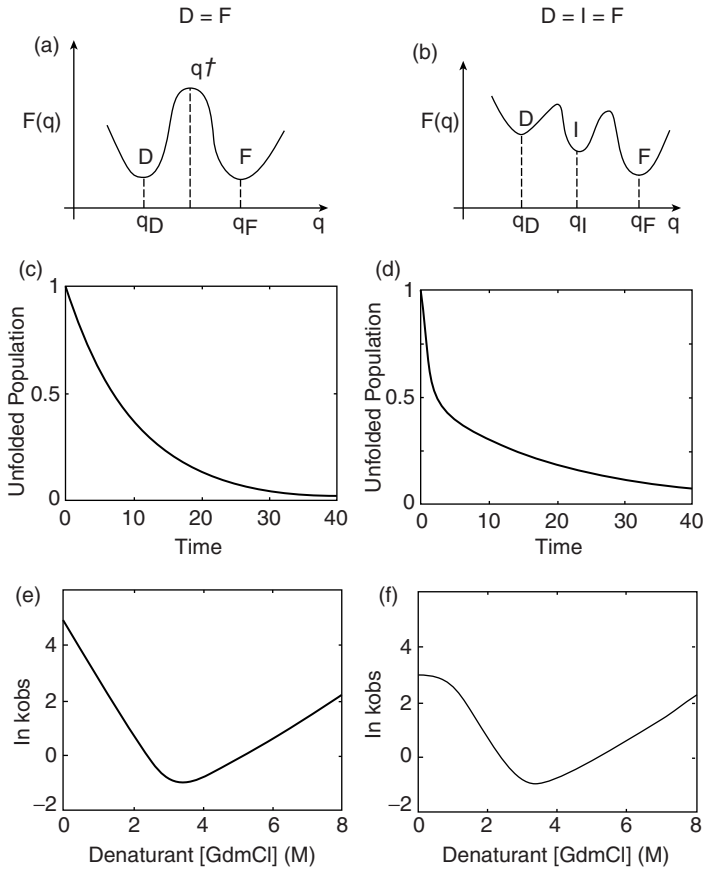


Figure 3 Two-state (D = F) versus three-state (D = I = F) folding. (a) Free energy surface for a two-state folder as a function of the reaction coordinate q. (b) Free energy surface for a three- state folder as a function of the reaction coordinate q. An additional minimum corresponding to an intermediate state I is present. (c) Single exponential kinetics of folding for a two-state folder. (d) Nonexponential kinetics of folding for a three-state protein. (e) Linear chevron plot for a two state folder. (f) Chevron plot with rollover for a three-state folder.

using a denaturant (typically urea or guanidinium chloride) and then dilutes the denaturant and monitors the refolding process using a spectroscopic signal (fluorescence, absorbance, etc.) that can distinguish the folded from the unfolded state. Data from these experiments are used to generate plots of the observed folding rate ($k_{obs} = k_f + k_u$) as a function of denaturant concentration. These plots are known as chevron plots because they are V-shaped, with "arms" corresponding to the folding rate k_f (at low denaturant concentration) and unfolding rate k_u (at high denaturant concentration). These

plots are linear for two-state folders but show a curve (or "roll over") when intermediates are present. Note that the absence of thermodynamic and kinetic signatures corresponding to intermediates does not preclude their existence; two-state behavior can be observed when high-energy intermediates are present.

In the next section, we present a tutorial on simplified and fully atomic simulation methodologies that can be used to investigate the features of energy landscapes as well as to reveal the nature of the thermodynamics and kinetics of folding.

PROTEIN MODELS

Introduction and General Simulation Techniques

As the computational resources available to the scientific community have improved, so too has our ability to use computational methods as a means for testing theories and for providing a link between theory and experiment. Protein folding is one field in particular where simulation has allowed access to information not available through experimentation. Simulations have been particularly useful with respect to testing the predictions of energy landscape theory. As mentioned in the Introduction to this chapter, three main types of models and techniques are commonly used by the protein folding community: lattice protein models via Monte Carlo simulations; off-lattice protein models with dynamics such as Langevin, Monte Carlo, and discontinuous molecular dynamics (DMD) methods; and atomically detailed simulation models that use both implicit and explicit solvent and that implement a variety of dynamics schemes. We now discuss these methods in more detail, beginning with lattice models and working our way up in complexity of computational methodology.

Although different simulation techniques are appropriate for different models and lines of inquiry, common concerns originate with respect to all of them. One concern of paramount importance is assessing the quality of the data that have been collected. For simulations to appropriately capture protein behavior at a particular temperature (or for a particular dielectric constant, solvent, or other parameter), they must sample available protein conformations adequately. To ensure that the conformations being sampled are representative of a particular parameter set, sampling must be performed over periods of time that allow the system to relax to thermodynamic equilibrium. Determining this equilibrium can be difficult, particularly if sampling is performed at or near a transition temperature.

One way of determining thermodynamic equilibrium is through the calculation of protein relaxation times. By choosing an appropriate variable to monitor folding progress (examples of which may be R_g, the radius of gyration, or Q, the number of native contacts formed) and monitoring its

autocorrelation function, one can determine the relevant relaxation time. In discrete form, the non-normalized autocorrelation function of a variable A is given by

$$C_{AA}(\tau) = \langle A(\tau)A(0)\rangle = \frac{1}{\tau_{\max}}\sum_{\tau_0=1}^{\tau_{\max}} A(\tau_0)A(\tau_0+\tau) \quad [1]$$

If A is saved at equally spaced time steps throughout the simulation, then $C_{AA}(\tau)$ can be calculated at any one of these points, labeled τ. As τ increases, $C_{AA}(\tau)$ should decay to 0 at τ_{rel}, because the value of A at long times should be uncorrelated with its initial value. To collect data for a system in thermodynamic equilibrium, the total simulation time must be 10 to 100 times longer than τ_{rel}. For a more complete discussion of this topic, as well as a good general discussion of block analysis, an alternative method for determining equilibrium sampling, we refer the reader to the classic book by Allen and Tildesley.[42]

Another concern common to all simulation techniques is the problem of "trapping" in local energy minima. Because proteins are complex biomolecules, they often have energy landscapes with many local energy minima particularly at low temperatures. Using standard simulation techniques, proteins often become trapped in these non-native state minima. Several techniques that have been developed to address this problem are known as generalized ensemble methods. These methods typically weight each state with a specific non-Boltzmann weighting factor (that is often difficult to determine). Because the probability for a protein to surmount the barrier between two local energy minima is exponential in 1/T, many high energy states are virtually inaccessible. By introducing a non-Boltzmann weighting factor, those states become accessible to the protein, which can then sample them and thus escape from energetic traps. Examples of generalized methods include multicanonical sampling, which is a random walk in energy space;[43] simulated tempering, a random walk in temperature space;[44,45] 1/k sampling, a random walk in entropy space;[46] J-walking;[47] use of the Tsallis ensemble;[48–50] the Wang Landau method;[51] and replica exchange (also known as parallel tempering).[52–54] An excellent review of generalized ensemble methods has been written by Mitsutake et al.[55]

After thermodynamic sampling at several temperatures has been obtained, the data from these simulations can be used to approximate a density of states for the system of interest. Once the density of states is known, other thermodynamic quantities may be calculated. Although it is possible to thoroughly sample a range of conformations at each temperature, the density of states that can be extracted from a single temperature is limited mostly to those energy levels having a high probability of being sampled. To obtain a more suitable density of states over a broad energy range, the weighted

histogram analysis method (WHAM) was proposed. First developed by Ferrenberg and Swendsen[56,57] and subsequently formalized for biomolecules by Kumar et al.,[58] WHAM provides a means for combining the data obtained at many different temperatures and generating a density of states that is more accurate than any one distribution obtained from just a single simulation at one temperature. It also allows for the calculation of thermodynamic quantities at temperatures other than those that were simulated.

Although our discussion has thus far centered on thermodynamic analyses, several issues originate with respect to kinetics. To calculate average folding times accurately, hundreds of simulations must be run at any given temperature. Although it is feasible for lattice and simple off-lattice models, fully atomic simulations cannot yet access time scales relevant to folding kinetics in a single simulation, which renders accurate folding time calculations impossible. Ergodicity is another concern, particularly for lattice-based simulation methods. The accessibility of any given state, starting from any other state, is not guaranteed for a given Monte Carlo move set. Additionally, folding times determined by lattice Monte Carlo simulations are dependent on the move set chosen. For molecular dynamics simulations, ergodicity is less of an issue, because the proteins are not required to move on a lattice. Even so, quantities like average folding times at high temperatures, or the fraction of unfolded proteins at a given temperature, depend on how long the simulation is allowed to run. Some of these issues will be discussed with respect to specific models below.

Coarse-Grained Protein Models

A goal of using simple computational models is to answer general questions about protein folding such as: What are the forces that drive the folding process? What makes proteins fundamentally different from random heteropolymers that lack a unique ground state? How do differences in energy landscapes translate to differences in folding? These questions are uniquely suited for a class of protein models called ''simple exact'' models[9,59–61] that use a reduced representation of amino acids and that are typically implemented to model short sequences (<40 residues). For very short sequences, one can enumerate completely all conformational states of a polypeptide. This in turn allows for the computation of an exact partition function and thus the derivation of relevant thermodynamic quantities. Simple exact models also have been used successfully to test the assumptions of analytical models, and they do not require approximations or assumptions beyond those inherent in the simple exact model. Additionally, many concepts gleaned from these simple exact models can be applied to more realistic protein models. Because we cannot discuss all simple exact models in detail here, we will focus on a few key models and the results derived from them and refer the reader to numerous references that discuss more thoroughly these and other models.

Lattice Models

Lattice models were first used to study proteins by Gō et al.[62,63] who used a hypothetical potential energy function that included only attractive contacts between two residues when those residues are in the native conformation. This potential is not realistic because it "forces" residues into the native conformation, nor is it a simple exact model; yet it provided initial insight into computationally appropriate ways to study protein folding. Additionally, the notion of "turning off" attractive interactions for non-native contacts is one that has elucidated many aspects of energy landscape theory, and Gō-type models are often used and referred to today.[64,65]

One of the earlier successful "simple exact" heteropolymer models is the HP model in both two and three dimensions.[66] This model has been studied at length by Chan and Dill.[67–71] The results of these studies have general applicability vis-a-vis understanding energy landscape theory and are still relevant when applied to more complicated theoretical models or experimental results. In the HP model, each amino acid is represented as a single bead that is either hydrophobic (H) or hydrophilic (P). Each bead is located on a lattice site, with the bonds connecting them represented as straight lines between nearest neighbors on the lattice. No lattice site may have more than one bead, which preserves excluded volume. A favorable contact free energy ε (where $\varepsilon < 0$) exists between HH pairs that are nearest neighbors on the lattice but not adjacent in the amino acid sequence. All other pair-wise combinations (HP, PP) contribute zero to the free energy of the peptide, so that the total energy for a given conformation is hε, where h is the number of HH contacts.

By using a simple two-dimensional model, one can explore both the conformational and the sequence spaces for a polymer of a given length. So, one can ask for a given HP sequence: What is the total number of conformations available to that peptide? Conversely, given a peptide of length n, one can ask the following: What are the possible combinations of HP sequences? For a particular sequence of length n, one can define g(h) as the density of states for a particular number of HH contacts, and

$$\Omega_0(n) = \sum_{h=0}^{h=h_N} g(h) \qquad [2]$$

as the total number of distinguishable states for that polymer. Here h_N is the number of HH contacts in the native state, which is designated as the most compact state that maximizes the number of HH contacts for a given sequence. For low enough n (usually $n \leq 22$), g(h) can be counted exactly for all h, and the partition function,

$$Q = \sum_{h=0}^{h_N} g(h) e^{-\varepsilon h/kT} \qquad [3]$$

can be computed. From Q, other relevant thermodynamic quantities can be determined. By comparing these quantities for different HP sequences of the same length, one can get a sense of how sequence impacts the thermodynamics of folding.

Several important thermodynamic results were obtained from analysis of the HP model. It was shown, for example, that as chain length increases, g(h) for low-energy compact states becomes independent of chain length, regardless of sequence.[72] Instead of a random search through conformational space, proteins search through a reduced number of compact states with low energy to find the native state. Thus, even though the total number of protein conformations increases with chain length, the ability of a protein to find its native state is not diminished.

Specific heat curves for a variety of HP sequences (where $C_v = \left(\frac{\partial U}{\partial T}\right)_V$) are shown in Figure 4 which depicts an example of both the sequence-dependent thermodynamics of the HP model and the rich thermodynamic behavior that can originate from such a simple model. Sequence (iii), which has a sharp peak in its specific heat curve (Figure 4b) and a sigmoidal distribution of its fractional native population (Figure 4a), is an example of a sequence with a gap between its native state and the ensemble of misfolded compact states. It has a well-defined hydrophobic core and a two-state folding behavior; at its folding temperature, only the unfolded and native states are significantly populated and there are no intermediates in the folding process. Sequence (i) has a significant number of hydrophobic residues on the outside of the protein. It undergoes a more gradual shift in its fractional native population and has a relatively broad specific heat curve with two small peaks; this example of a sequence does not fold in an all-or-nothing transition. Sequence (ii) has a more intermediate behavior.

To explore the kinetics of a polymer with the HP model, an appropriate set of moves on the lattice (move set) is critical, given the concerns mentioned earlier. In their study of the HP model, Chan and Dill[71] used the two different move sets shown in Figure 5a and b. For each move set, they constructed the appropriate adjacency matrix; two conformations are considered adjacent if one can be transformed into the other by a single move in the move set. Although they found that the kinetics of folding is strongly dependent on both sequence and move set, several universal features of folding were uncovered. First, chains can fold through multiple paths, some of which include significant kinetic traps. Second, proteins tend to get stuck in collapsed, low-energy, non-native conformations, and they must overcome energetic barriers by unfolding or rearranging to get to the native state. Third, the sequences that fold fastest are those with an energy gap between their native state and the conformation with the next highest energy level. These results are consistent with other work, particularly that of Shakhnovich et al.[73,74] In their early work on lattice models with two types of beads, Shakhnovich et al. used a sequence annealing procedure to determine the lowest energy sequence (with

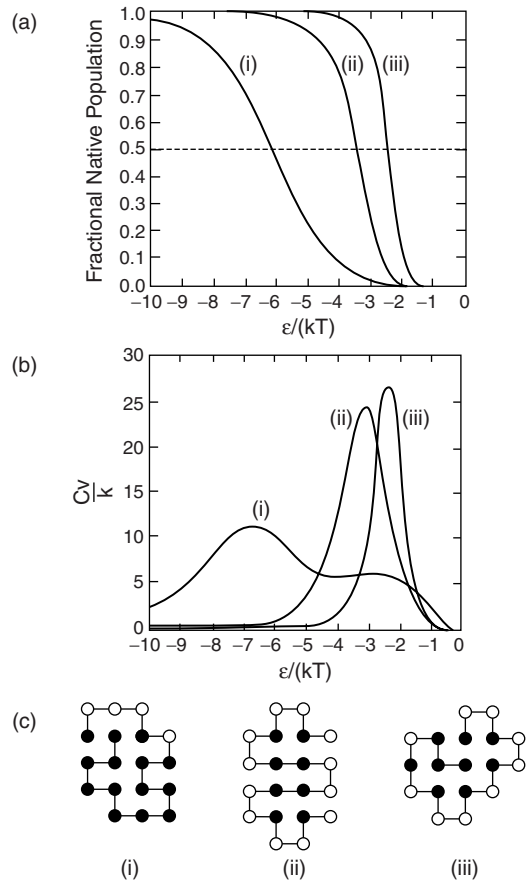


Figure 4 Three different two-dimensional HP sequences (panel c) and their corresponding specific heat (panel b) and fractional native state population (panel a) curves. (Adapted from Dill et al.[9])

a fixed ratio of amino acids) for a given structure. Using these selected sequences, they could draw conclusions about the folding of designed versus random sequences.

In addition to the HP model, several other low-resolution protein models have been developed to provide insight into the nature of protein folding. The AB model, used in the studies mentioned by Shakhnovich et al. and studied extensively by Socci and Onuchic,[60,75] is similar in spirit to the HP model. The AB model is a three-dimensional lattice protein with 27 monomers. Each monomer is either of type A or type B, and the interaction energy between two nearest-neighbor, nonbonded monomers is E_ℓ for an AA or BB pair, and E_u for an AB pair, where $E_\ell < E_u < 0$. In this scheme, like contacts are favored over unlike contacts, and there is an overall driving energy toward

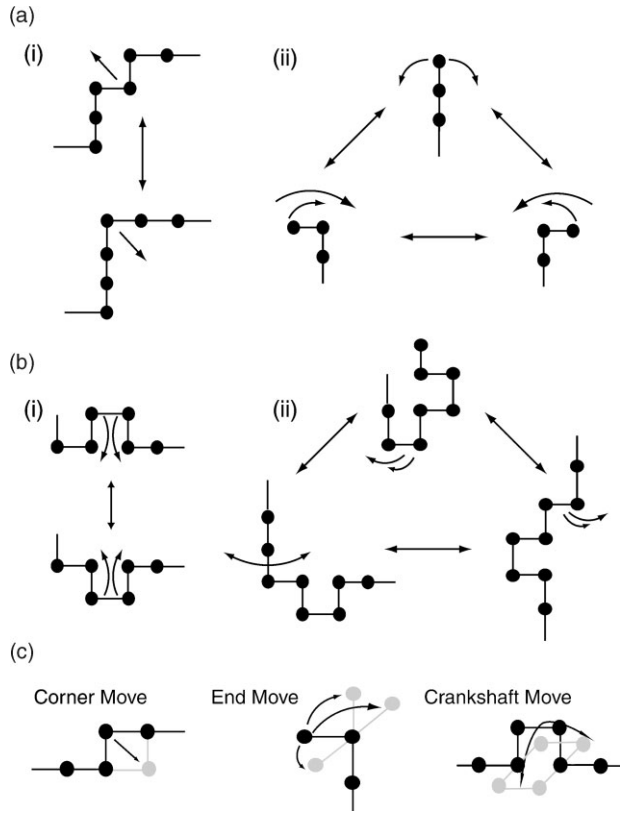


Figure 5 Examples of lattice move sets. Shown (a) and (b) are the two different move sets used in the simulations performed by Chan and Dill.[71] The double-ended arrows represent adjacent conformations, and the single-ended arrows indicate the residues being moved. In move set (a) only one residue is moved per Monte Carlo step; the selected residue may be (i) moved diagonally as indicated (three bead flip), or (ii) an end residue may be pivoted about its neighbor (end flip). In move set (b), two or more residues may be moved in a single Monte Carlo step. In (i), the residues undergo a crankshaft move, whereas (ii) is a rigid rotation. (c), a Monte Carlo move set used by Socci and Onuchic.[75] The black circles indicate the current position of the residues on the lattice, with the gray residues representing their future positions. The corner move, like the three bead flip employed by Chan and Dill, moves a single residue on the corner of the lattice to a diagonally adjacent position. The end move rotates an end residue around its neighbor to one of three possible adjacent locations, and the crankshaft move flips two residues by 90 degrees. This move set also demonstrates the three-dimensional generalization of some elements in the move sets used by Chan and Dill.

collapse, because any contact is favored over no contact. The total energy of a given conformation is thus $E = N_\ell E_\ell + N_u E_u$, where N_ℓ and N_u are the number of like and unlike contacts, respectively. It is not feasible to enumerate all conformations available to this 27 monomer lattice protein. However, all

maximally compact conformations, where the protein resides on a $3 \times 3 \times 3$ lattice, can be counted.

The move set used for both thermodynamic sampling and dynamics (the use of the word dynamics here should be taken cautiously) of the AB model is similar to that employed for the HP model. The move set includes corner, end, and crankshaft moves as indicated in Figure 5c. To ''advance'' from one conformation to the next, a monomer is chosen at random, and, depending on its location on the lattice, either a corner, an end, or a crankshaft move is attempted. If the attempted move puts the monomer on an already occupied site, that move is rejected to preserve excluded volume, and a new monomer is chosen. Once a monomer has been selected and the move deemed possible, the energy of that new conformation is calculated. If the new conformer's energy is lower than that of the previous conformation, the move is accepted. If the energy is higher, it is accepted with the usual Boltzmann probability:

$$P = \exp[-(E_{new} - E_{old})/T] \qquad [4]$$

Here, Boltzmann's constant is set equal to 1. Regardless of whether a move is accepted or rejected, one unit of time (one Monte Carlo step) is considered to have passed. This probabilistic acceptance criterion is known as the Metropolis Monte Carlo algorithm.[19] Although no connection exists between physically relevant time scales and Monte Carlo time steps, Monte Carlo simulations can estimate the relative time scales of protein folding versus simulation time, as well as the time needed to reach equilibrium at a given temperature. Keep in mind, however, that any time scale extracted from a Monte Carlo simulation depends on the move set used. Even so, useful information can be extracted from such a simulation, such as relative transition times for two different sequences.

In their analysis of six different 27-residue AB sequences, Socci and Onuchic could reveal a variety of thermodynamic and kinetic behaviors. They examined both the compaction time (number of steps to reach a collapsed state with 25 of 28 native contacts formed) and the folding time (time from a random, unfolded to native state) for all six sequences at various temperatures. They found that the compaction time is sequence independent, whereas the folding time is highly dependent on sequence. From their analysis of compaction and folding times, Socci and Onuchic could conclude that the proteins they studied exhibit a two-phase folding mechanism. First, the protein would collapse to a low-energy compact state in a time that is roughly sequence independent. Second, that stage was followed by one where the protein searches for its minimum energy state from these low-energy collapsed states, a process that is highly sequence dependent. Another result from this work is that proteins that folded the fastest had low native state energies and high folding temperatures. Those sequences with T_f below the glass transition temperature T_g did not reach their native state quickly at T_f, a temperature

at which they are thermodynamically stable. Additionally, T_g was found to be sequence independent. Thus, even with a model as simple as the AB model, a key prediction of energy landscape theory was tested and verified. More specifically, proteins that fold on a biologically relevant time scale must have $T_g < T_f$; otherwise, the protein can become trapped on a rugged energy landscape. Socci and Onuchic used WHAM to do a thermodynamic analysis. They could look at quantities such as the heat capacity C_V and the probability distribution of the energy P(E) for several sequences at a variety of temperatures. They found that the collapse transition is second order-like in nature, with the folding transition being more abrupt and first order-like. Those sequences that folded fastest and were most stable in their folded states had sharper and more distinct thermodynamic transitions. Not coincidentally, these sequences lacked energetic frustration in their native states; that is, all energetic contacts made were favorable contacts.

The lattice models mentioned above were among the first to test the predictions of energy landscape theory as well as to show that many important features in protein folding could be captured by simple physical principles. In other studies, lattice models have been extended to include side-chain elements,[61,76,77] side-chain-only models,[78–80] and diamond, body-centered cubic (bcc) and face-centered cubic (fcc) lattices. These models, while maintaining their simplicity and computational tractability, are much more realistic than those discussed above, and will likely provide further insight into the interactions that drive protein folding. For an excellent review on reduced protein models, we refer the reader to Kolinski and Skolnick and the extensive references therein.[81]

Off-Lattice Models

Off-lattice minimalist models are similar to the lattice models in that they generally use a simplified amino acid representation. Rather than being confined to a lattice, the protein is free to move in continuous space. As with lattice models, many different off-lattice models have been studied. Some are meant to be reduced models of specific proteins,[31,33,82] whereas others are meant to capture a specific secondary structure motif, such as an α-helix, a β-sheet, or an α–β sandwich (see citations in Ref. 81). As before, we will focus on a few representative models and provide appropriate references about the others.

One of the first off-lattice minimalist models was developed by Honeycutt and Thirumalai (HT) in 1990 to model a β-barrel motif.[83] It models each amino acid as a single bead like those previously described, but it incorporates much more realistic energetic interactions by accounting for both nearest- and non-nearest-neighbor forces as well as for bond and torsion angles. The HT model includes three different types of residues: hydrophobic (B), hydrophilic (L), and neutral (N). Consider the sequence for this 46-residue protein: $B_9N_3(LB)_4N_3B_9N_3(LB)_4L$. Its Hamiltonian is as follows.

Interactions between residues separated by three or more bonds in the sequence are dependent on residue type. For two hydrophobic residues, the energy between them is given by a Lennard–Jones potential:

$$4\varepsilon_h\left[\left(\frac{\sigma}{r_{ij}}\right)^{12}-\left(\frac{\sigma}{r_{ij}}\right)^{6}\right] \qquad [5]$$

where σ is the diameter of each residue and r_{ij} is the distance between the two residues in consideration. An LL or BL interaction has the following form:

$$4\varepsilon_L\left[\left(\frac{\sigma}{r_{ij}}\right)^{12}+\left(\frac{\sigma}{r_{ij}}\right)^{6}\right] \qquad [6]$$

and the interaction of an N residue with any other type of residue is given by $4\varepsilon_h(\frac{\sigma}{r_{ij}})^{12}$. Note that $\varepsilon_L = \frac{2}{3}\varepsilon_h$. To find the total interaction, we sum over all pairs of residues that are separated by three or more bonds. To model the torsion and bond angles, the following functions are used:

$$\sum_{\substack{\text{bond}\\ \text{angles}}}\frac{1}{2}k_\theta(\theta-\theta_0)^2+\sum_{\substack{\text{torsion}\\ \text{angles}}}(A(1+\cos\varphi)+B(1+\cos 3\varphi)) \qquad [7]$$

with adjustable parameters A and B set by the type of residues involved in the interaction. The bonds between nearest-neighbor residues are kept fixed. The native state of the HT model is shown in Figure 6a.

The dynamics scheme used for the HT model is different from those previously discussed. Here, Langevin dynamics is used. The Langevin equation for a single particle with position x_i is given by

$$m\ddot{\boldsymbol{x}}_i = F(\boldsymbol{x}(t)) - \gamma v_i(t) + \Gamma(t) \qquad [8]$$

and includes three terms on the right-hand side. The first term is the force on the residue caused by all other residues [hence the notation $F(x(t))$ to indicate that F does not depend solely on x_i]. The second term is a damping function to systematically model solvent viscosity. The third term models random noise, such as the thermal kicks on the protein from the solvent. The magnitude of $\Gamma(t)$ is related to the damping term through the fluctuation dissipation theorem, where $\langle\Gamma(t)\rangle = 0$,

$$\langle\Gamma(t)\Gamma(t')\rangle = c\delta(t-t') \qquad [9]$$

and $c = 2\gamma k_B T$. Each residue of the protein is subject to the constraints of the Langevin equation, and the equations of motion are integrated with the velocity form of the Verlet algorithm.[84]

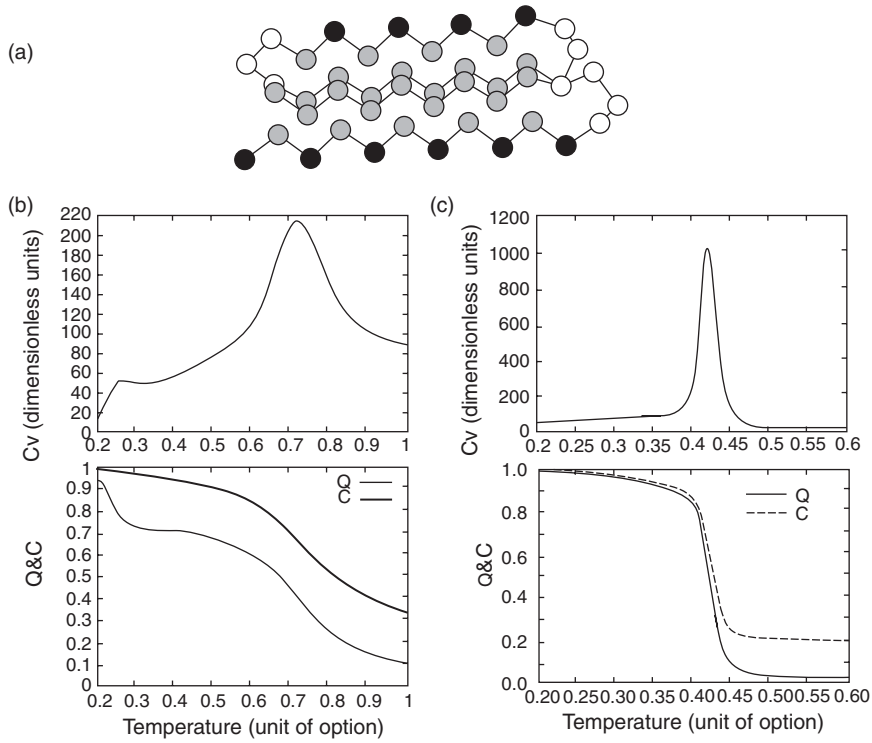


Figure 6 (a) The native state of the HT model. Hydrophobic residues are black, hydrophilic residues are gray, and neutral residues are white. (b) Specific heat (C_v) curve and plots of the average fraction of native contacts (Q) and total number of contacts (C) as a function of temperature for the original HT model. (c) Same curves as in (b) for the minimally frustrated Gō-model. The original HT model shows a nonspecific transition, whereas the minimally frustrated model displays the signatures of a two-state folder. Adapted From Nymeyer et al. (Ref. 64).

The original studies using the HT model[83–85] focused on exploring the nature of its native state and probing the kinetics of folding. Honeycutt and Thirumalai found that many low-energy conformers are populated under folding conditions. The protein was observed to fold rapidly to one of these compact intermediate states, but the time scale for the protein to find its true native state could be extremely long. Their results indicate that it is possible for the rate-limiting step in protein folding to occur late in the folding process. In this case, the rate-limiting step is caused by a local rearrangement of contacts within the protein, rather than by large-scale fluctuations. Folding was observed to be nonexponential, further confirming a noncooperative folding mechanism. Subsequent studies using this model[86] revealed a three-state folding mechanism involving collapse from the unfolded state to a compact misfolded state, followed by a transition to the native state. In a later study, Guo and Brooks[87] probed the thermodynamics of protein folding with the HT model in even

more detail. Using WHAM,[58] they identified two relevant thermodynamic transition temperatures for the folding process: the collapse temperature T_θ, and the folding temperature T_f. At T_θ, the protein collapses from an extended state to a compact but non-native low-energy structure, and at T_f, it goes from this collapsed state to the native state.

The kinetic and thermodynamic results obtained from the HT model are indicative of a rugged energy landscape. This landscape comes from the non-specific character of the hydrophobic interactions, which lead to the formation of compact, albeit incorrectly formed structures. Onuchic, Brooks, and co-workers[32,64,88] developed less frustrated versions of the HT model by introducing specificity into the hydrophobic interactions. Nymeyer et al.[64] constructed a Gō-like version of the HT model by identifying all native contacts (defined as contacts in the native state within a distance of 1.167σ), and then turning off the attractive part of the Lennard–Jones interaction for all but those contacts. The result is a model in which only the native contacts are favored. The authors compared the specific heat and fraction of native contacts formed for both the original HT model and their new, nonfrustrated model. The original model has a broad specific heat curve (Figure 6b), whereas the Gō-like Nymeyer model has a narrowly peaked specific heat curve (Figure 6c), indicating a sharp transition. Furthermore, plots of the average fraction of native contacts and the total number of contacts as a function of temperature change in a concerted manner in the Gō-like model, which implies concomitant folding and collapse ($T_f = T_\theta$). In contrast to the original model, folding with the Gō-like Nymeyer model was observed to be single exponential at temperatures below T_f, another signature of a two-state cooperative folding mechanism. Nymeyer et al. estimated the glass transition temperature for these two models and found that the T_f/T_g ratio was about 0.9 for the original model and near 8 for their Gō-like model, which is consistent with criteria for frustrated and unfrustrated proteins, respectively, as given by energy landscape theory.

The HT model was one of the first successful off-lattice models, and many others have been developed since then. Models are now being used with two or more atoms per residue,[89–93] with all-atom Gō-models (Gō- interactions with atomically detailed protein model),[94,95] and with the addition of charge[87] and solvation effects.[96–99] These models are being systematically improved to better capture realistic protein interactions, while remaining simple enough for large-scale, long-time simulations to uncover the underlying physics behind protein folding.

Fully Atomic Simulations

The most thorough representation of folding would involve describing both the protein and its environment in explicit atomic detail. This level of detail comes at a significant computational cost, with typical simulations consisting of hundreds of protein atoms and thousands of water molecules.

Systems of this size prohibit the use of a computational approach based solely on quantum mechanics, a method that can only provide an exact solution for systems with limited number of electrons. To study the dynamics of proteins, an approximate solution to the Schrödinger equations, such as the one given by molecular dynamics using force fields, is required. This type of molecular dynamics is based on three approximations:

1. The Born–Oppenheimer approximation that allows decoupling of the motions of the electrons and the nuclei.
2. A treatment of the nuclei as classic particles moving on a potential energy surface (PES). Trajectories of the nuclei on the PES are obtained by numerically solving Newton's equations of motion: $F_i = -\nabla_i U = \frac{d^2 r_i}{dt^2}$, where F_i is the force acting on particle i, U is the potential energy, r is the particle position, and t is time. Several schemes can be used to numerically integrate Newton's equations of motion, including the Verlet and Leap-Frog algorithms.[100]
3. An approximation of the potential energy surface by a potential energy function (PEF) describing the physical interactions between the particles. The PEF permits the calculation of the potential energy and interatomic forces as a function of the coordinates of the system. In the case of proteins, the PEF are atom-based rather than nuclei-based.

A typical PEF used in biomolecular simulations consists of bonded and nonbonded interaction terms:

$$U(R) = U_{bd}(R) + U_{nonbd}(R) \qquad [10]$$

with the bonded contribution $U_{bd}(R)$ consisting typically of bond length, bond angle, and torsion terms:

$$U_{bd}(R) = \sum_{bonds} k_b \Big(b(R) - b_{eq}\Big)^2 + \sum_{angles} k_\theta \Big(\theta(R) - \theta_{eq}\Big)^2 + \sum_{dihedrals} k_\phi \Big[1 + \cos\Big(n\phi(R) - \gamma\Big)\Big] \qquad [11]$$

where b, θ, and ϕ are the bond lengths, bond angles, and dihedral angles, respectively, and k_b, k_θ, and k_ϕ are the associated force constants. The subscript "*eq*" denotes equilibrium values. The value n in the torsional term corresponds to the periodicity and γ to the phase. The nonbonded term $U_{nonbd}(R)$ consists typically of van der Waals (Lennard–Jones term) and electrostatic components:

$$U_{nonbd}(R) = \sum_{\substack{nonbonded \\ atom\ pairs\ i,j}} \left(\varepsilon_{ij} \left[\left(\frac{R_{\min,ij}}{r_{ij}} \right)^{12} - \left(\frac{R_{\min,ij}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{\varepsilon_D r_{ij}} \right) \qquad [12]$$

where ε_{ij} is the well depth and $R_{\min,ij}$ is the distance between atoms i and j at which the Lennard–Jones interaction is at a minimum. The terms q_i and q_j correspond to the charges of atoms i and j, r_{ij} to the distance between the atoms, and ε_D to the dielectric constant. The various parameters in the PEF are obtained from experiments and quantum mechanical calculations. Force fields that are common in biomolecular simulations such as CHARMM,[101] AMBER,[102,103] OPLS,[104] and GROMOS[105] consist of the PEF and their corresponding parameters. Accuracy in force field parameterization is important for obtaining realistic trajectories and energies from simulations.

As mentioned, a serious computational obstacle associated with fully atomic, solvated simulations involves the many particles needed to represent the system. United atom models, in which the hydrogen atoms of the protein are considered to be part of the heavy atoms to which they are bound, reduces the size of the system, but true, atomic-level detail is lost. The real bottleneck in simulation time is not the solute but instead the thousands of water molecules needed to properly solvate the protein. Most computer time is spent calculating water molecules' interactions with one another, whereas the dynamics of the protein are really the elements of interest to most researchers. The presence of water molecules greatly increases the size of the system, and it requires averaging over long times to obtain meaningful simulation results for the protein. Early simulations on protein folding attempted to reduce the computational burden associated with water molecules by performing the simulations in a vacuum. As one might expect, it provided a misleading picture of folding because electrostatic terms are overemphasized without a polar solvent and the critical hydrophobic effect is neglected. More recent, although still crude approaches, involve using distance-dependent dielectrics in an attempt to treat the solvent. A more sophisticated "implicit solvent" model common in folding simulations is the generalized Born (GB) model, based on an approximation of the Poisson–Boltzmann equations.[106] When coupled with a solvent-accessible surface area (SASA) term, the GB/SASA models provide a satisfactory description of both electrostatic and hydrophobic effects, at a much reduced computational cost when compared with using explicit water molecules.[107,108]

Another computational limitation associated with molecular dynamics simulations on fully atomic systems is the need to use small time steps when integrating Newton's equations of motion. To maintain an accurate and stable simulation, the time step must be smaller than the fastest motions of the system, which in this case involve vibrations of the heavy atoms—hydrogen atoms' bonds (X-H stretch). Integration time steps of 1 fs (2 fs if algorithms such as SHAKE[109] are used to fix the X-H bonds) are commonly employed, which means that one million integration time steps are required to generate a single nanosecond trajectory. As a result of this limitation, most folding simulations have been restricted to nanosecond time scales. Sampling of conformational space is hence limited and studying the kinetics of folding, which

occurs on times scales of microseconds to minutes, becomes prohibitively costly. An example of a long-time, fully atomic molecular dynamics simulation on a protein in explicit solvent is the 1-μs simulation performed by Duan and Kollman on the villin headpiece, a 36-residue helical protein with an estimated folding time of 10–100 μs.[110,111] Folding was initiated from an unfolded conformation and reached a metastable, compact state bearing some similarity to the experimentally determined native state within 4 months of simulation time on a CRAY supercomputer with 256 processors. Although being a computational "tour de force," a single simulation of this type cannot provide the statistics needed to evaluate folding kinetics and thermodynamics; methods other than such simple "brute force" techniques are required to gain insight into the folding process.

We review below several methods aimed at overcoming the computational obstacles associated with protein folding simulations. Because these methods have been applied to many protein systems, too numerous to be discussed in detail in this review, we focus instead on a few specific proteins that have been studied by more than one method. Of particular interest is the folding of Fragment B of Protein A (Figure 7). This protein has served as a model system for theoretical studies involving both minimalist[31,82] and fully atomic protein models.[112–116] Recent experiments by Sato et al.[117] have put these simulations to the test, with the conclusion that although simulations have been able to identify some main elements of this protein's folding, none of them have been able to provide a picture completely consistent with experiment. It highlights the limitations associated with force fields, water molecules, simulation issues (sampling, etc.), and so on, which are facts that must be kept in mind when drawing conclusions from simulation data. The following salient experimental results exist for protein A. First, folding is "two-state" and occurs on the time scale of milliseconds, with Helix III forming

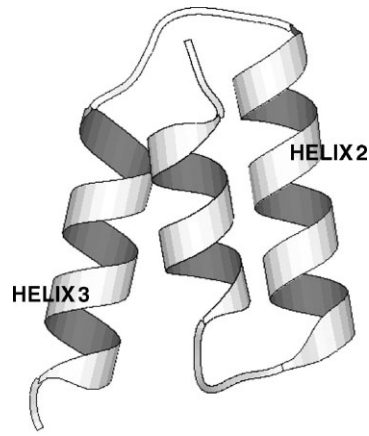


Figure 7 Ribbon diagram of fragment B of protein A.

early in the folding process and being the most stable structural element in isolation.[118,119] Second, based on protein engineering experiments by Sato et al.,[117] folding seems to follow a nucleation-condensation mechanism with a transition state for folding in which Helix III (the most stable as a fragment) is poorly formed, particularly in the C-terminal region. Third, folding seems to be initiated around a well-structured Helix II, with partial formation of the Helix II–Helix III turn and with little structure in the Helix I–Helix II turn region. We will return to these experimental observations throughout much of the chapter.

Stochastic Difference Equation (SDE) Method

Elber et al. introduced a method, based on a functional formulation of classical mechanics, that allows for large time steps by using an integration scheme with respect to path length rather than time.[120] Their SDE generates trajectories for long-time processes, such as protein folding, that are inaccessible using classical molecular dynamics methods. Rather than solving Newton's equations of motion, the action is optimized, with the goal of finding a trajectory that makes the action S stationary. The action is given by

$$S = \int_{Y_u}^{Y_f} \sqrt{2(E-U)}\, dl \qquad [13]$$

where Y_u and Y_f are the mass-weighted coordinates of the unfolded and folded states, E is the total energy, U is the potential energy, and dl is the mass-weighted length element of the system. The associated equations of motion are

$$\frac{d^2Y}{dl^2} = \frac{-(\nabla U - (\nabla U.e)e)}{2(E-U)} \qquad [14]$$

where e is the unit vector. In discretized form, the action is given by

$$S \cong \sum_i \sqrt{2(E-U)}\Delta l_{i,i+1} = \sum_i \left(\left(\frac{\Delta^2 Y}{\Delta l^2} \right) + \left[\frac{dU}{dY} - \left(\frac{dU}{dY} \right) e \right] e \right)^2 \qquad [15]$$

Trajectories are obtained by minimizing the gradient norm:

$$T = \sum_i \left(\frac{\partial S / \partial Y_i}{\Delta l_{i,i+1}} \right)^2 \Delta l_{i,i+1} + \lambda \sum_i (\Delta l_{i,i+1} - \langle \Delta l \rangle)^2 \qquad [16]$$

which yields the exact classic trajectory for small steps. The last term in Eq. [16] is a penalty function ensuring that all points are spread out along the trajectory evenly.

The SDE method has been applied to the folding of two proteins, the 46-residue helical fragment B of protein A[116] and the large 104-residue Cytochrome C.[121] Both proteins were modeled in atomic detail using the AMBER/OPLS united atom force field[103,104] and with an implicit GB/SA solvent model.[122] The SDE method does not yield equilibrium-free energy surfaces for folding. It does, however, allow for a study of the direct sequence of events connecting an unfolded conformation to the folded state.

We focus here on Protein A, in which the following folding mechanism emerges from the SDE simulations. First, the unfolded state consists of an extended conformation with a few contacts present in Helix III near the C-terminus. Second, compaction of the protein is accompanied by formation of the secondary structure. Finally, tertiary contacts form concurrently with the remaining secondary structural elements. Folding is observed to obey neither the "hydrophobic collapse" model nor a model in which substantial secondary structure forms early on. The folding process is depicted in Figure 8 and presented as plots of $(-RT \ln\{P(Rg, N_{hb})\})$ as a function of the radius of gyration Rg and the number of native hydrogen bonds N_{hb}. $P(Rg, N_{hb})$ is the joint probability of the radius of gyration and the number of native hydrogen bonds averaged over 130 trajectories.

Protein Unfolding

The millisecond time scales associated with protein folding under native conditions and protein unfolding under mild denaturing conditions prohibit their study with a straightforward molecular dynamics approach. Using an elevated temperature (400 K and greater) would accelerate reaction rates by several orders of magnitude, from milliseconds to nanoseconds. It is tempting to use unfolding trajectories to infer the folding mechanism. However, caution must be used when making such inferences. The principle of microscopic reversibility, which implies that unfolding mechanisms are the reverse of folding mechanisms, applies only under equilibrium conditions and is not expected to hold when extreme perturbations are applied to the system. Simulations performed on simplified lattice and off-lattice models have shown that unfolding pathways obtained under harsh denaturation conditions are quantitatively different from folding pathways generated under native conditions.[18,123] Nonetheless, unfolding pathways often appear qualitatively to be the reverse of folding pathways, as the secondary elements that form last also tend to be the first to be disrupted. Unfolding simulations are likely to be most useful when studying the folding of proteins with highly polarized folding mechanisms, such as for the SH3 protein domain[124] and CI2.[125–127]

Levitt et al. used high-temperature (498 K) simulations to study the unfolding of fragment B of protein A using the ENCAD force field[128] and with an explicit treatment of solvent. Their simulations indicate that the protein adopts a compact denatured state within a few nanoseconds, with Helix III being the last element to unfold. Daggett et al. infer that Helix III

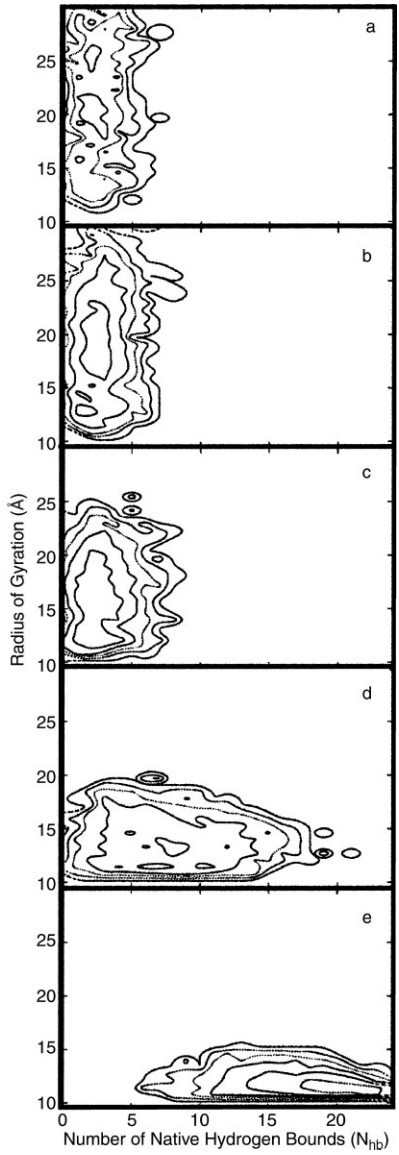


Figure 8 Sequence of folding events from plots of $(-RT \ln\{P(Rg, N_{hb})\})$ as a function of the Rg radius of gyration and the N_{hb} number of native hydrogen bonds at different times along the trajectory. $P(Rg, N_{hb})$ is the joint probability of the radius of gyration and the number of native hydrogen bonds that in this case have been averaged over 130 trajectories. (a)–(c), correspond to the early stages of folding and show a small increase in the number of hydrogen bonds with gradual compaction of the protein. It is followed in time (d) a large change that occurs in both Rg and N_{hb}, indicating collapse with formation of secondary structure. The final stages of folding (e) involve further formation of hydrogen bonds with little change in Rg. Adapted from Ghosh et al.[116]

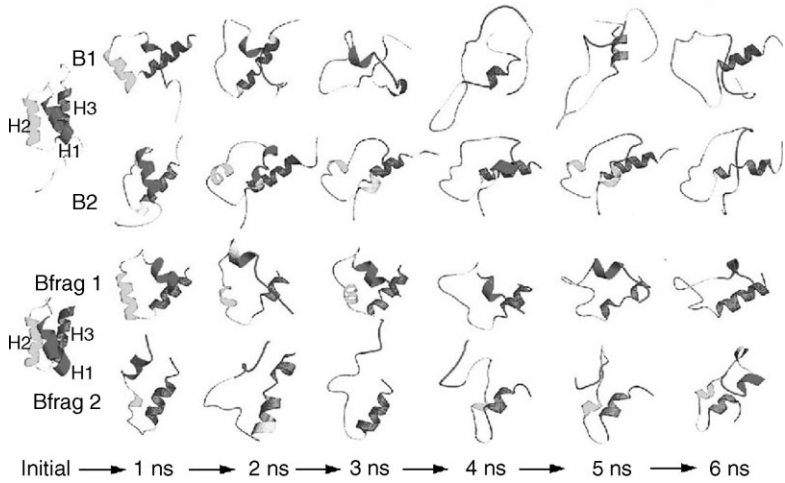


Figure 9 Unfolding simulations of the full-length protein (from two runs labeled B1 and B2) as well as a 46-residue fragment (from two runs labeled Bfrag1 and Bfrag2). The full-length protein and the fragment unfold in a similar manner, with Helix III losing structure last. Adapted from Alonso and Daggett.[114]

is the most stable structural element and likely the first to form on the folding pathway. Helix III acts as a ''scaffold'' around which the other two helices form. The transition state is found to be compact, with 63% of native contacts formed. Helix III is the most structured element of the transition state, with Helices I and II structured to a lesser extent. The unfolding trajectories of this protein, as well as those of a 46-residue fragment, are shown in Figure 9. A discussion of the method used to determine transition states from unfolding simulations is given in the section ''Advanced Topics: The Transition State Ensemble for Folding.''

Importance Sampling Molecular Dynamics

An elegant means to generate free energy surfaces for folding was developed by Brooks et al.[18,112] The method involves denaturing simulations and the use of hierarchical clustering algorithms to identify representative conformations spanning the folded to the unfolded states. These conformations serve as initial conditions for importance sampling, which is performed under refolding conditions with a harmonic biasing potential along a reaction coordinate (typically a continuous analog of the number of native contacts). Data for the sampling runs are then combined with a constant temperature version of the WHAM algorithm, which yields the density of states as a function of temperature and reaction coordinates. The free energy surface (potential of mean force) is obtained from the density of states. The method is efficient because each initial condition is independent of others. The method remains

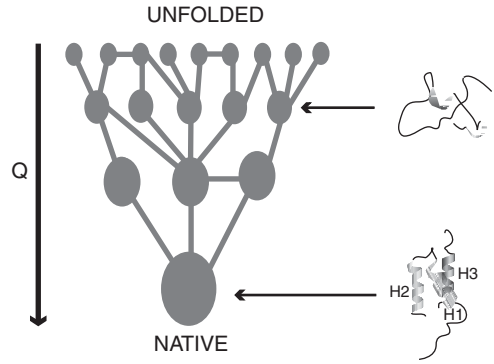


Figure 10 Sampling scheme in the importance sampling method. Each ellipse represents a conformation used in biased sampling. The lines indicate the connected network created through sampling. Adapted from Guo et al.[113] with protein structures from Alonso and Daggett.[114]

costly in terms of time as sufficient sampling must be generated to create a connected net between the initial conditions (see Figure 10). The importance sampling methodology was first applied to the helical fragment B of protein A and subsequently to several other proteins, including the mostly β-sheet proteins G[129] and src-SH3.[18,130–132] Simulations were performed with the CHARMM force field[101] and with explicit TIP3P water molecules. Free energy surfaces were generated as a function of the radius of gyration, the number of native contacts, and the number of hydrogen bonds. These simulations identified different characteristics of the free energy surfaces for helical and β-sheet proteins. Helical proteins were observed to have a surface with a diagonal shape, which is consistent with concomitant collapse and folding, whereas β-sheet proteins showed a more "L-shaped" profile, indicative of an initial collapse before folding. The simulations also revealed the critical role of water in the final stages of folding. The free energy surfaces of both protein G and SH3 have a compact, near-native solvated basin separated by a small energy barrier from the native basin. In the case of protein G, this solvated basin corresponds to structures in which water molecules intercalate between misregistered β-sheets. These water molecules serve as "lubricants" that facilitate proper alignment of the sheets. In the case of SH3, the solvated basin consists of conformations with water residing between the two hydrophobic sheets constituting the hydrophobic core of the protein. The final folding event for SH3 involves the expulsion of water molecules to form a "dry" core.[130,131]

The free energy surfaces generated for fragment B of protein A portray the following picture for folding. Folding is initiated at the Helix I–Helix II turn, which is followed by formation of stable secondary helical structures

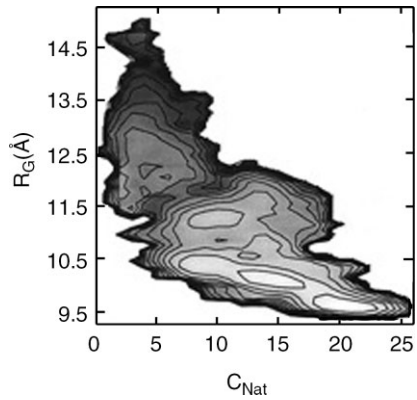


Figure 11 Free energy surface as a function of the radius of gyration and the number of native contacts for protein A from an importance sampling molecular dynamic simulation. Adapted from Guo et al.[113]

in Helices I and II. The final folding step involves formation of Helix III, which becomes stable only when making contact with Helices I and II. The transition state, which is determined from the barrier on the free energy surface (plotted as a function of the radius of gyration and the number of native contacts), is compact, with 30% of all native state contacts present and 50–70% of the native state hydrogen bonds formed. Hydrogen bonds in Helix II have the highest probability of forming, whereas those in Helix III are less probable. Transition state structures resemble an expanded form of the native structure, with overall correct topology. The free energy surface for protein A is represented as a function of the radius of gyration and the number of native contacts in Figure 11.

Enhanced Sampling Methods: Replica Exchange Molecular Dynamics

Conventional molecular dynamics methodology applied to biological molecules tends to result in an incomplete sampling of conformational space. It leads to a distorted statistical picture of the conformational ensembles populated under a given set of conditions and to incorrect conclusions regarding both folding mechanisms and protein conformational preferences. Several enhanced sampling schemes have been developed recently to remedy this sampling problem by facilitating escape from local energy minima.[55,133] One of the most promising methods is the replica exchange (REX) algorithm, which was introduced initially in the context of spin glasses.[54] Details of the REX formulation for molecular dynamics can be found in the seminal paper by Sugita and Okamoto.[134] In this scheme, several identical copies, or "replicas," of the original system are simulated, in parallel, for a given number of MD steps at different temperatures. Two replicas i and j adjacent in temperatures

T_i and T_j, with energies E_i and E_j, are swapped periodically with the probability:

$$p_{ij} = \begin{cases} 1 & for \quad \Delta \leq 0 \\ \exp(-\Delta) & for \quad \Delta > 0 \end{cases} \qquad [17]$$

where $\Delta \equiv [(\beta_i - \beta_j)(E_j - E_i)]$ and $\beta = 1/k_B T$.

Because the escape time from local energy minima decreases significantly at elevated temperatures, the REX method enables both enhanced equilibration and sampling by treating the temperature as a dynamical variable. In addition to leading to a more thorough exploration of conformational space, the algorithm also ensures that the conformations sampled at a given temperature in the REX simulations belong to the canonical statistical ensemble. It allows for application of ensemble reweighting techniques[58] to extract equilibrium thermodynamic functions, including, for instance, the mean potential energy as a function of temperature.

REX has emerged as an increasingly attractive and tractable means to study the thermodynamics of folding of peptides and small proteins in the last few years.[135–137] We focus here on the replica exchange molecular dynamics (REMD) investigations of the folding of fragment B of protein A done by Garcia and Onuchic[115] who used the AMBER force field[102] with explicit solvent molecules. The enhanced sampling protocol allowed for the generation of free energy surfaces at temperatures above and below the folding transition temperature. The free energies were decomposed into their enthalpic and entropic contributions, both of which were observed to change in a "downhill" manner with an increasing number of native contacts Q. The incomplete cancellation of enthalpy and entropy led to a barrier in the free energy surface between the unfolded and the folded states. Folding was observed to be an overall two-state process, in agreement with experimental findings. Interestingly, the folded basin at low temperature consists of two basins separated by a very small barrier. Those two minima correspond to a folded state with a hydrated core and to fully desolvated folded conformations, which highlights the role of water desolvation in the final stages of protein folding. The free energy profiles for folding under native conditions and at the transition temperature are represented in Figure 12 as a function of the root-mean-squared deviation (RMSD) and the number of native contacts Q. The simulations reveal a folding mechanism involving an interplay between secondary structure formation, desolvation, and formation of the hydrophobic core. The unfolded state is found to be compact, with Helix III forming early. The transition state shows significant structure in Helix I, with tertiary interactions between Helices I–II and Helices II–III and to a lesser extent between Helices I–III. Helix I seems to be unstable in the absence of interactions with the other two helices. Despite forming early in the unfolded state, Helix III does not interact fully with the other helices until the later stages of folding. The overall

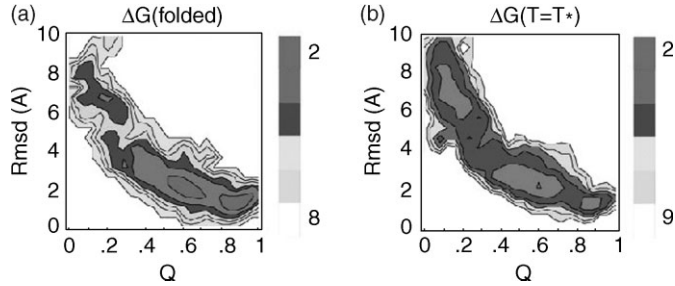


Figure 12 Free energy surfaces for protein A computed with REMD simulations. (a) shows the surface as a function of RMSD and the number of native contacts below the folding transition temperature. The native state shows two basins corresponding to a hydrated nearly folded state and to the ''dry'' folded state. (b) shows the surface as a function of RMSD and the number of native contacts at the folding transition temperature. The unfolded and folded states are equally populated at this temperature. Taken with permission from Garcia and Onuchic.[115]

folding landscape is found to be funneled, with a certain overlaying frustration evident from the presence of non-native contacts in the folding simulations.

The simulation methods presented in this section demonstrate the range of approaches developed to tackle the computational challenges associated with simulations of folding processes. Despite limitations intrinsic to each method, along with force field and water model shortcomings, these simulations have been able to provide insight into folding at a level of detail not accessible to experiments.

ADVANCED TOPICS: THE TRANSITION STATE ENSEMBLE FOR FOLDING

In this section on advanced folding topics, we provide a broad overview of computational approaches used to determine the transition state ensemble (TSE) for folding. Comprehensive reviews are available elsewhere on closely related topics, such as experimental characterization of the transition states[138,139] and free energy landscape approaches to the transition states.[38]

Three groups of methods used to determine protein transition states for two-state folders will be covered. The first group involves approaches that use one- or multidimensional reaction coordinates for folding to locate transition states. The second group uses rigorous theoretical approaches that do not rely on the introduction of any specific reaction coordinate.[65,140–142] The third group of methods uses information from experimentally measured ϕ-values (parameters that quantify the relative changes in folding rates and stability of the native state of a protein induced by point mutations) as input to reconstruct the transition state structures.[143–145]

Transition State and Two-State Kinetics

Most small, single-domain proteins display thermodynamic and kinetic signatures of two-state folders. The simplicity of this folding scenario, in which only unfolded (U) and native (N) states are populated to any significant degree, led to the development of kinetic models analogous to those first introduced in the context of chemical rate processes.

We begin by reviewing models for chemical reactions of small molecules and then extend this formalism to folding reactions. The description of a rate process in terms of reaction models is based on two premises: (1) the existence of thermodynamically stable reactant and product states that can interconvert and (2) the existence of a unique reaction coordinate that can distinguish between these two states. A typical potential energy surface $U(q)$ for a two-state rate process involving the conversion between a reactant A and its product B is given in Figure 13. States A and B are the only thermodynamically stable states in this reaction and appear as minima on the energy surface, separated by a barrier. Transitions between states A and B can be described fully by the dynamics of a one-dimensional reaction coordinate q. The time for a transition to occur is dominated by the time required to overcome the energy barrier located at $q^{\dagger}$, with a descent into the other energy minimum occurring on a much faster time scale. According to transition state theory (TST),[146] the ensemble kinetics of the reaction schematically represented in Figure 13 is single-exponential with the reaction rate $k = 1/\tau$ given by an expression:

$$k = k_T \exp(-\Delta U/k_B T) \qquad [18]$$

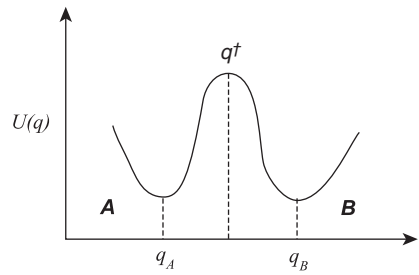


Figure 13 A typical one-dimensional potential energy surface for a chemical reaction. Two stable states corresponding to the reactants *A* and the products *B* are separated by a barrier. The transition between the two states is described by a reaction coordinate q that takes on different values for the reactants and products. The rate-limiting step corresponds to reaching the conformation located at the top of the barrier $q^{\dagger}$, which is the transition state. Depending on the nature of the underlying dynamical process, the rate of the reaction can be predicted accurately by one transition state theory or the Kramers formalism.[146] In protein folding, the reactants *A* are associated with the unfolded state and the products *B* are taken to be the folded native state. The reaction coordinate typically remains unspecified.[65]

where $\Delta U = U(q^{\dagger}) - U(q_A)$ is the barrier that must be overcome in the $A \rightarrow B$ direction of the reaction, k_B is Boltzmann's constant, T is the temperature, and k_T is the transmission coefficient (i.e., the time required for barrierless transition). A similar expression for the transition time can be obtained under certain conditions using a Kramer's type approach.[7] Because the rate-limiting step in the reaction is reaching the state $q^{\dagger}$ located at the top of the barrier (the transition state), an accurate characterization of this state is key to describing reaction kinetics.

An extension of the reaction-rate formalism to protein folding is not straightforward, and requires more than a simple assignment of the unfolded state to the reactants and the folded state to the products. The main stumbling blocks are the protein's many degrees of freedom and the intrinsic difficulty in determining which dynamical variable is a suitable reaction coordinate for folding. In contrast to the simple one-dimensional case considered in our previous example, one needs to make a critical distinction between *order parameters* used to identify thermodynamically stable reactants and products and *reaction coordinates* that can be used to predict the transition time between these two states. Indeed, not all good order parameters serve as equally good reaction coordinates.[148] Modeling a protein folding reaction requires knowledge of both a reaction coordinate and its dimensionality. Another fundamental difference between small molecule reactions and protein folding is that the latter requires a statistical description. As mentioned, above protein folding thermodynamics and kinetics are described in terms of ensembles rather than single microstates. The transition state is hence a TSE, consisting of a collection of conformations that can be structurally diverse.

In the case of a one-dimensional process (Figure 13), knowledge of the reaction coordinate allows one to unambiguously assign the transition state as the state at the top of the energy barrier. For reactions taking place in higher dimensions, no straightforward relationship exists between reaction coordinate and transition state. In principle, a proper reaction coordinate for a complex biomolecular reaction exists when the dynamics of a single, primary degree of freedom occurs on a much slower time scale than the dynamics of its secondary degrees of freedom. The secondary degrees of freedom can then be accounted for implicitly, through statistical averaging, projection onto the primary variable, and introduction of a friction term to model their dynamics. The resulting reduction of configuration space simplifies the problem greatly, and the TSE can be predicted accurately from a one-dimensional reaction coordinate. The problem with this approach is that for most biomolecular reactions, one does not know a priori if a one-dimensional reaction coordinate exists. Recent simulations using on-lattice[149] and off-lattice[147] protein systems lead to conflicting results regarding whether such a reaction coordinate can truly be found. More recent work[151] on continuous protein models indicates that initial guesses for reaction coordinates can be improved through rotations in the phase space of available order parameters. This work shows that the

one-dimensional formalism of chemical kinetics can lead to a quantitatively accurate description of folding kinetics and thermodynamics when a carefully selected reaction coordinate is used. Several successful applications of the one-dimensional approximation for the reaction coordinate of folding have been reviewed recently by Kubelka et al.[150] Although a one-dimensional reaction coordinate may be a suitable approximation for the folding reaction, the true dimensionality of the folding reaction coordinate is unknown and a general treatment of folding must account for its possible multidimensionality.

The theoretical foundation for defining a transition state in multidimensional space is based on the concept of a *stochastic separatrix*. The stochastic separatrix is defined as the locus of structures having equal probabilities of reaching the products (native state) and reactants (denatured state). This idea was first introduced in the context of condensed-phase chemical reactions[152] and later applied to the protein folding problem by Du et al.,[65] who introduced the *commitment probability* function P_{fold}. P_{fold} is defined as the probability for a protein configuration to reach the native state before reaching the unfolded state. It is a kinetic parameter with a specific value for a given protein conformation, which enables identification of a particular structure's location in the multidimensional configurational space. The transition state is defined as the ensemble of conformations with $P_{fold} \approx 1/2$. For continuous protein models, the commitment probability is evaluated by folding simulations, with hundreds of simulations launched for each initial structure. The fraction of runs that reach the native state before reaching the unfolded state is taken to be P_{fold}. This quantity is costly to compute numerically because an exhaustive sampling of the entire configuration space with subsequent evaluation of P_{fold} is required to determine the TSE accurately. Because of this high computational cost, exact computations of TSE have been limited to simplified lattice proteins[65,123,153] and to small peptides in vacuum.[148] The P_{fold} method most frequently tests the quality of putative reaction coordinates.[94,148,154–157] Note that just because a TSE can be determined from the set of conformations with $P_{fold} \approx 1/2$, one is not guaranteed that a true one-dimensional reaction coordinate for folding has been identified. Furthermore, because the P_{fold} method is not based on geometrical parameters for folding, it is difficult to extract a physically meaningful reaction coordinate from the TSE, even if such a coordinate does indeed exist.

Methods for Identifying the TSE

In this section we describe the computational methods available for extracting transition state conformations through computer simulations. Three classes of methods are discussed: (1) those relying on the introduction of one or more reaction coordinates, (2) rigorous methods not linked to any specific reaction coordinate, and (3) methods based on the protein engineering ϕ-value formalism.

Reaction Coordinate-Based Methods

Reaction coordinate-based methods identify TSE structures as those conformations residing at the top of the free energy barrier projected onto a given reaction coordinate q (see Figure 13).[65] Several different reaction coordinates have been introduced to extract TSE conformations from simulations,[31,124,158,159] with the most common coordinate being the number of native contacts Q.[59] A serious concern with reaction coordinate-based approaches is that the putative reaction coordinate may be a poor descriptor of the folding process. Although Q seems to be a suitable reaction coordinate for Gō-type models, it is often an inadequate reaction coordinate for proteins with a higher degree of frustration[160] such as the Cα-Cβ model of the C-src SH3 domain[157] and in a model three-helix bundle recently studied by Baumketner et al.[151] In a simulation of chymotrypsin inhibitor 2 (CI2),[161] Q failed to distinguish between pre- and post-transition conformations. Identifying a simple geometric parameter that will act as a good reaction coordinate[65] is a nontrivial task. The failure of any given reaction coordinate to properly identify the TSE seems to be the rule rather than the exception. When an inadequate reaction coordinate is used, the ensemble of conformations at the top of the free energy barrier will include non-TSE conformations in addition to true TSE structures as illustrated in Figure 14. Equally worrisome is that members of the true TSE that do not have $q = q^{\dagger}$ will not be identified from free energy surfaces projected onto a poor reaction coordinate q, and that at best only part of the TSE can be obtained from reaction coordinate-based methods.

An improvement over methods that use a one-dimensional reaction coordinate for locating transition states is to expand the space of relevant reaction coordinates into multiple dimensions. An example of such an approach is the conformational clustering method of Li and Daggett[125] in which the TSE is determined on the basis of structural changes that occur as the protein unfolds.[162] Structural fluctuations are categorized as being local or large scale, and transition state structures are defined as the ensemble of structures

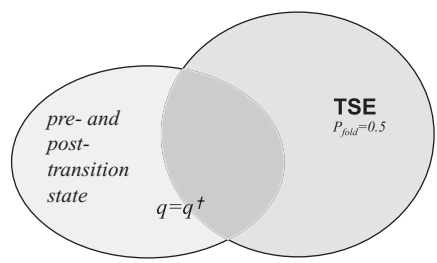


Figure 14 Overlap of the TSE determined on the basis of a nonideal reaction coordinate q with the true TSE. The reaction coordinate-based TSE contains both pretransition state ($P_{fold} < 0.5$) and post-transition state ($P_{fold} > 0.5$) conformations and omits some true TSE conformations ($P_{fold} = 0.5$).

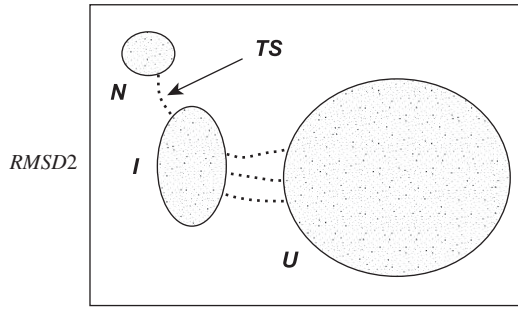


Figure 15 Identifying the transition state conformations from unfolding simulations under denaturing conditions. Trajectories are projected onto the two-dimensional space spanned by two RMSD components and analyzed to determine regions of highest residence. RMSD1 and RMSD2 are defined such that the geometrical distance in 2D space between structures i and j closely matches the actual RMSD between these structures. The regions of highest residence are associated with the thermodynamically stable states *N* and *U* and possibly intermediates *I*. Transition states in this method[125] correspond to the conformations found immediately after the protein leaves the native state basin in a trajectory.

populated immediately before the onset of a large structural change. To determine the TSE, unfolding trajectories are projected onto a two-dimensional space constructed from the RMSD among all conformations recorded in the simulation (see schematic depiction in Figure 15). Thermodynamically stable states correspond to the regions of highest residence in this projection. The major macroscopic states *N*, *U*, and (possibly) *I* are identified, and those conformations found immediately after the trajectory leaves the native state basin are selected as transition states. It is equivalent to identifying the transition state conformations as the least-visited states in the two-dimensional space defined by the chosen reaction coordinates (here RMSDs). It should be noted that this method is not rigorous unless the folding reaction is two-dimensional and the selected RMSDs constitute the correct reaction coordinates of folding. Extensive numerical tests have been performed to validate the use of the method. The method has been shown to adequately locate transition state structures in certain instances. In particular, the putative transition state structures determined by this method were found to have commitment probabilities P_{fold} close to one half in simulations of chymotrypsin inhibitor 2 (CI2).[163] In other words, for this protein, the method could indeed locate transition state structures correctly.

A potential limitation of Li and Daggett's method[125] for identifying folding transition state structures is its reliance on unfolding simulations. In general, one can expect that the unfolding process, which occurs under conditions where the protein is unstable, follows different kinetic pathways than for folding, which occurs under physiological conditions. Consequently, folding

and unfolding transition states are not necessarily the same. The free energy landscape theory of protein folding predicts that the TSE is influenced by environmental factors such as the temperature or the concentration of denaturant. Using Monte Carlo simulations of a lattice protein model, Dinner and Karplus[123] observed (in accordance with energy landscape theory) that the unfolding TSE depends on the degree of denaturation. In addition, folding and unfolding TSEs, determined using the more rigorous P_{fold} formalism,[65] were different. The transition state for unfolding at a temperature above the melting temperature seemed in that latter study to be more structured than the transition state for folding below the melting temperature. In view of this finding, it is imperative to delineate accurately the conditions under which unfolding simulations can be used to map out folding TSEs.

Nonreaction Coordinate-Based Methods

The method developed by Thirumalai and coworkers[141,164,165] is a rigorous method that does not use reaction coordinates to locate transition state structures. This method attempts to detect a folding nucleus through folding simulations. Nucleation is a kinetic phenomenon satisfying the following requirements: (1) The folding nucleus consists of some minimal number of stable native contacts in which stability is determined by the probability of breaking a contact (native or otherwise) over some specified amount of time. (2) The formation of the nucleus is the rate-limiting step of the folding reaction and is followed by a rapid assembly of the native state.

The numerical implementation of this method involves first generating many (typically 100 or more) folding trajectories at temperatures where the native state is stable. Each trajectory is characterized by a first passage time τ_i that is determined using the structural overlap parameter $\chi(t)$ as a measure of nativeness of the protein. The trajectories are then analyzed in terms of their probability for forming native contacts. A contact q, formed at some time before τ_i, is taken to be stable if it remains unbroken before the native conformation is reached. A histogram of stable contacts (contact map) originating at some time during the simulation for a typical trajectory i is shown in Figure 16. The periods of time when stable contacts are formed are indicated by thick lines. It is clear from the plot that (1) most contacts do not form until very late in the folding process (time $\delta \cdot \tau_i$ in Figure 16) and (2) rapid assembly of the native state is triggered by the formation of a minimal set of native contacts. These two observations form the conceptual basis of Thirumalai's method for identifying folding nuclei. The numerical implementation of this method introduces a variable δ (scaled time) for defining the moments in time at which folding nuclei are formed. For a fixed δ, a function $P_N(q)$, corresponding to the probability that contact q is formed at time $\delta \cdot \tau_i$ and remains stable until the first passage time τ_i, is constructed by averaging over all recorded trajectories. In general, the $P_N(q)$ for various contacts are insignificant until relatively high δ values (between 0.8 and 0.9) are reached.

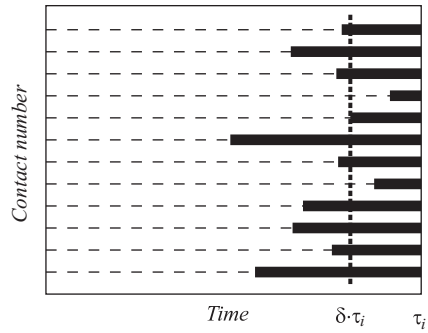


Figure 16 A typical time evolution of native contacts plotted for a given trajectory i versus first passage time τ_i. Thick lines indicate periods of time when stable native contacts are formed. Stability means that once formed, a contact q remains intact until the native state is reached. A critical nucleus is searched for in conformations recorded at times $[\delta \cdot \tau_i, \tau_i]$ (see the text for a more detailed explanation of how δ is chosen).

A minimal set of native contacts with the highest formation probability is designated as a critical nucleus for each studied trajectory i. Because this assignment of critical contacts is dependent on the specific values of δ used for evaluating $P_N(q)$, a numerical procedure is introduced for computing δ_{TS} that unambiguously defines transition state structures. This procedure examines a test function $P(\delta)$ obtained from $P_N(q)$ by averaging over all native contacts q. The derivative of $P(\delta)$ with respect to δ is taken, and δ_{TS} is identified as the time when a rapid growth in $dP(\delta)/d\delta$ occurs. It is assumed that each trajectory crosses the transition state at time $\delta_{TS} \cdot \tau_i$. Conformations belonging to this state are identified by clustering all conformations recorded at times $[\delta_{TS} \cdot \tau_i, \tau_i]$.

Using lattice models of proteins, Klimov and Thirumalai tested the accuracy of their method for identifying transition state conformations[154] by computing a commitment probability P_{fold} for each putative transition state conformation they located. For states generated from $\delta_{TS} = 0.9$, an average $P_{fold} \sim 0.56$ was observed for one sequence studied, which leads the authors to conclude that their method of generating the transition state is equivalent (numerically) to the stochastic separatrix approach. This claim should be viewed with caution, however. To unambiguously determine whether these conformations belong to the TSE, an examination of both the mean values of P_{fold} as well as their distributions for the given set of conformations is needed. Indeed, cases in which a significant population of pretransition state ($P_{fold} < 0.5$) and post-transition state ($P_{fold} > 0.5$) conformations are present (with an average P_{fold} close to 0.5) have been reported.[166]

Although rigorous in nature, Thirumalai's method is numerically costly even for the simplest Gō model, because it requires a significant number of folding trajectories to be generated. Because of this limitation, the method has not been used extensively. Systems studied to date include an off-lattice

β-barrel protein model,[141] a C_α-model of the terminal β-hairpin of the GB1 protein,[167] and lattice models.[154,164] An important conclusion derived from this set of studies is that multiple folding nuclei exist that enable folding via parallel pathways.

A second rigorous nonreaction coordinate method for identifying the transition state was developed by Dokholyan et al.[142] In contrast to the method of Guo and Thirumalai,[141] where explicit folding simulations are run from an ensemble of initial states, this method starts by generating a single, lengthy, constant-temperature trajectory under thermodynamic conditions in which the native state is stable. Multiple folding/unfolding events are observed and analyzed from this simulation.

A subset of folding/unfolding trajectories passing through a putative transition state are selected from the equilibrated trajectory. The putative transition state is identified on the basis of some trial reaction coordinate, such as the potential energy (that trial reaction coordinate is not used at any other point in the analysis). When searching for the critical nucleus, it is assumed that equilibrium fluctuations in the system can be divided into local and global types. Local fluctuations do not lead to a change of the thermodynamically stable (folded or unfolded) states, whereas global fluctuations correspond to the actual folding or unfolding of the protein from a denatured or native initial states.

Under native conditions, large local unfolding fluctuations will include the critical nucleus, whereas large local folding fluctuations will include the folding nucleus under denaturing conditions. Depending on the initial and final states of the simulation (U or N), the trajectories are separated into four classes: UU corresponds to pathways that start from the unfolded state (U) and never fold, NN to those that start from the native state (N) but never unfold, NU to those that start as folded proteins and end in unfolded states, and UN to the trajectory class for which folding is observed (see Figure 17). These classes of conformations behave differently with respect to the

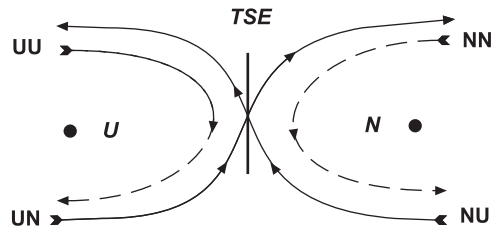


Figure 17 Sample folding pathways that pass (or may pass) through the transition state. The pathways are divided into four classes: UU are those paths that start from the unfolded state (U) and never fold, NN are those that start from the native state (N) but never unfold, NU are those that start as folded proteins and end as unfolded, and UN are those for which folding is observed. The frequency of contact formation is monitored. Contacts for which the difference between NN and UU classes is a maximum constitute the CN.

frequency of contact formation. NN trajectories are more likely to contain a critical nucleus than are trajectories for the UU states. A critical nucleus (if it exists) begins to form in UN pathways and breaks apart in the NU trajectories. Thus, the frequency of observing nucleation contacts for the UN and NU classes will lie in between the frequency of appearance for NN and UU classes. The difference in probability of contact formation for NN versus UU trajectories is used to locate the critical nucleus. The frequency of contact formation f_{UU} is computed for all contacts over UU trajectories and subtracted from a similar function for the NN ensemble. Depending on the relative population of contacts, it is expected that this differential function $f_{NN} - f_{UU}$ will take on values from some finite-range interval with well-defined boundaries $(f_{NN} - f_{UU})_{\min}$ and $(f_{NN} - f_{UU})_{\max}$. It is indeed the case for the C_α–C_β-based Gō model of the C-Src SH3-domain protein studied by Shakhnovich et al.,[142,157] where the largest difference $(f_{NN} - f_{UU})_{\max} = 0.2$. Native contacts with the maximal differential appearance frequency are taken to define those conformations for which the critical nucleus is formed. For the SH3 protein, only five specific contacts contributed to $(f_{NN} - f_{UU})_{\max}$. Proof that those five contacts belong to the critical nucleus was provided by simulations in which each of those contacts were fixed in order. In contrast to the bimodal energy distributions obtained in control simulations with a fixed random contact, the simulations with a fixed nucleation contact produced unimodal energy distributions, which indicates that the protein was never able to unfold, as it could not cross the barrier separating folded and unfolded states. This observation highlights the important role of a critical nucleus in the folding process.

An alternative definition of the transition state (not based on the commitment probability) was recently introduced by Hummer[140] who quantifies the difference between *equilibrium* distribution functions and the distribution functions obtained over *transition paths* (TP). Two regions corresponding to the reactants A and products B are defined in the entire configuration space. A path in this space is assumed to be reactive (transition path) if it starts from A and reaches B without returning to A. A conditional probability $P(TP|x)$ of being on a transition path, given that the system is at point x of the configuration space, is introduced. This probability is related to the equilibrium distribution $P_{eq}(x)$ and the conditional distribution $P(x|TP)$ of x over transition paths by Eq. [19]:

$$P(TP|x) = \frac{P(x|TP)P(TP)}{P_{eq}(x)} \qquad [19]$$

Here $P(TP)$ denotes the probability of the trajectory being on a transition path. $P(TP)$ is a multiplication factor that does not depend on x and reflects the fraction of time spent by a protein on transition paths over a long equilibrium trajectory in which multiple transitions have occurred. The transition state structures are defined as those points in the configuration space

corresponding to the maximum of $P(TP|x)$; transition states maximize the probability that folding paths passing through them are reactive. Note that Eq. [19] combines information from both equilibrium and transition path sampling.

For the case in which the configuration space is one-dimensional, Eq. [19] can be solved analytically using Langevin dynamics in the high-friction (diffusive) limit. It can be shown in this simple case that

$$P(TP|x) = 2P_{fold}(x)[1 - P_{fold}(x)] \qquad [20]$$

where $P_{fold}(x)$ is the usual commitment probability for point x (probability to reach products B in the general case). Because $P(TP|x)$ is quadratic in P_{fold}, it reaches its maximum value of 1/2 when $P_{fold}(x) = 1/2$. Recalling that Eq. [20] delineates the region in configuration space where points have equal probability of reaching either state A or state B, it becomes clear that in the one-dimensional case, the Hummer approach to identifying the TSE is equivalent to the method based on the stochastic separatrix idea described. The analytical expression for $P(TP|x)$ can be extended into multidimensional spaces using a shooting algorithm of transition path sampling.[168] Although slightly more complicated than in the one-dimensional case, the expression is based solely on commitment probabilities P_{fold} and thus predicts that points belonging to the stochastic separatrix will maximize $P(TP|x)$.

A generalization of the Hummer method that is useful for protein folding problems results when the entire configuration space is projected onto a selected low-dimensional (reaction) coordinate q. In this case, all equations derived above remain the same, with the distributions now pertaining to a subset of configuration space given by the condition $q = q(x)$. For example, the equilibrium distribution featured in the denominator of Eq. [19] is written as

$$P_{eq}(q) = \int \delta[q - q(x)]P_{eq}(x)dx \qquad [21]$$

It follows from Eq. [19] that $P(TP|q)$ is large when $q(x)$ is visited frequently in the transition paths and only rarely visited in equilibrium. The condition of being visited infrequently in equilibrium delineates the region of least probable points in configuration space. It corresponds to the free energy barrier. It thus follows that an appropriately defined reaction coordinate q should have a sharply peaked $P(TP|q)$.

The definition of transition states proposed by Hummer[140] allows for numerical implementations in computer simulations. $P(TP|q)$ can be determined conveniently by launching (''shooting'') trajectories from the phase points x on the surface $q(x) = q$,

$$P(TP|q) \approx \frac{\#\ \textit{accepted shooting moves at } q}{\#\ \textit{attempted shooting moves at } q} | P_{eq}(q) \qquad [22]$$

where x is drawn from the equilibrium distribution $P_{eq}(q) \sim P_{eq}(x)\delta(q - q(x))$. Alternatively, Eq. [19] can be used directly. Distributions $P_{eq}(q)$ and $P(q|TP)$ can be determined from generalized-ensemble techniques (such as multicanonical[43] or umbrella sampling[169]) and transition-path sampling methods,[168,170] respectively. Another option for computing these distributions would be to use a long and sufficiently equilibrated trajectory. The relative merits of these numerical methods have yet to be established, and no protein system has thus far been studied using Hummer's method.[140]

Determination of Transition States Based on ϕ-Value Analysis

ϕ-value analysis is a protein engineering technique introduced by Fersht et al.[171] to determine transition state structures through site-directed mutagenesis. A ϕ-value reflects the extent to which the transition state is destabilized by a mutation. It is defined as

$$\phi_T = \frac{\Delta\Delta G^{\dagger}}{\Delta\Delta G_0} \qquad [23]$$

where

$$\Delta\Delta G^{\dagger} = \Delta\Delta G^{\dagger}_{mut} - \Delta\Delta G^{\dagger}_{wt} \qquad [24]$$

is the change in the height of the folding free energy barrier and $\Delta G^{\dagger} = G^{\dagger} - G_U$, and $G^{\dagger}$ and G_U are the free energy of the transition and the unfolded state, respectively. The subscript T in Eq. [23] reflects the thermodynamic nature of this definition of ϕ, in contrast to a kinetic definition that we will introduce later in the text. The subscripts *mut* and *wt* in Eq. [24] denote the mutant and wild-type proteins. $\Delta\Delta G_0 = G_N - G_U$ reflects the change in the protein's stability after a point mutation has been made. G_N is the free energy of the folded native state, and G_u is the free energy of the unfolded state. Figure 18 illustrates all terms involved in Eq. [23]. The free energy barrier $\Delta G^{\dagger}$ cannot be measured experimentally in a direct way. However, for two-state folders (which possess single exponential kinetics), $\Delta G^{\dagger} = G^{\dagger} - G_U$ is related to the folding rate through

$$k = k_T \exp\left(-\Delta G^{\dagger}/k_B T\right) \qquad [25]$$

where k_B is Boltzmann's constant, T is the temperature, and k_T is the transmission coefficient that denotes the time required for a barrierless folding transition. Microscopic expressions for k_T depend on the theoretical description being employed. Several transition state theories, as well as Kramers' model of diffusive dynamics, suggest that the relationship given in Eq. [25] is the functional dependence of the folding rate on the free energy barrier.[146]

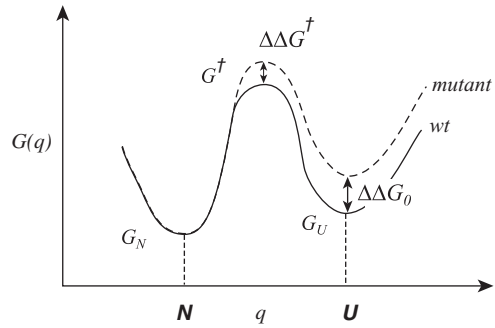


Figure 18 The various free energy terms involved in ϕ-value analysis.[171] The free energy as a function of a reaction coordinate q is plotted for the wild-type (wt) (solid line) and a mutated protein (dashed line). Mutations can affect both the stability of the native state $\Delta G_0 = G_N - G_U$ and the height of the folding free energy barrier $\Delta G^{\dagger} = G^{\dagger} - G_U$. The relative change in these quantities $\Delta\Delta G^{\dagger}/\Delta\Delta G_0$ upon mutation is the ϕ-value.

When the influence of mutations on the transmission coefficient are negligible, Eqs. [23] and [25] lead to the Eq. [26], which can be used experimentally to compute ϕ-values:

$$\phi_k = \frac{-k_B T \ln(k_{mut}/k_{wt})}{\Delta\Delta G_0} \qquad [26]$$

In Eq. [26], k_{mut} and k_{wt} are the folding rates of the mutant and wild-type proteins, respectively. The experimental ϕ-values bear the subscript k to denote their kinetic nature and to distinguish them from the thermodynamics ϕ_T values. ϕ-values measure the extent of structure in the TSE. Values close to 0 indicate that the mutation has little effect on the kinetics of folding ($\boldsymbol{k_{mut} \approx k_{wt}}$) and that the mutated residue lacks structure in the TSE. ϕ-values close to 1 correspond to mutations that affect folding rates (and hence the free energy barrier for folding) dramatically and have mutated residues that experience a native-like environment in the TSE. We emphasize that Eq. [26] is valid only when the protein displays single exponential kinetics and when the preexponential factor in Eq. [25] is invariant to protein mutation. We note that it is possible to devise protein models having exponential folding kinetics but with a folding rate that is not described by Eq. [25]. Examples of such systems are proteins with low free energy barriers, where folding is described by a full Kramers' expression[7] but not by its approximate form.[172] Given these caveats, it is desirable to investigate in greater depth the conditions under which the kinetic ϕ_K and thermodynamic ϕ_T values are equivalent. Understanding these conditions is especially important because it can help determine when ϕ-value analysis can be used to characterize protein transition states.

Lattice simulations by Onuchic et al.[172] reveal that the relationship between thermodynamic and kinetic ϕ-values depends strongly on the

frustration of the protein model. For unfrustrated models, a near-perfect match between ϕ_k and ϕ_T parameters exists, whereas frustrated sequences display a poor correlation between ϕ_k and ϕ_T. A similar conclusion about the role of frustration in determining the validity of Eq. [26] was reached by Shea et al.[32] It was shown that large amounts of frustration lead to a clear deterioration of the agreement between the thermodynamic and the kinetic ϕ-values using simulations of an off-lattice protein model. Moreover, it was found that sequences containing moderate amounts of energetic frustration may still possess two-state folding kinetics. Despite this apparent two-state behavior, the correlation between ϕ_k and ϕ_T was found to be poor for these sequences. Nonexponential kinetics, a signature of excessive amounts of frustration in a protein, rules out the use of ϕ-values for determining TSE structures.

ϕ-values can be evaluated numerically in several ways. ϕ_k-values are determined from Eq. [23] by running kinetic simulations (of the wild-type and mutated proteins) and determining the respective folding times/rates.[32,167,172] Free energy differences $\Delta\Delta G_0$ are estimated using a folding order parameter that can distinguish between the denatured and the folded states (this parameter need not be a successful reaction coordinate). Thermodynamic ϕ_T-values are determined numerically by reexpressing Eq. [23] as follows:[31,33,157]

$$\phi = \frac{\log\langle e^{-\beta\Delta E}\rangle_{\dagger} - \log\langle e^{-\beta\Delta E}\rangle_{U}}{\log\langle e^{-\beta\Delta E}\rangle_{N} - \log\langle e^{-\beta\Delta E}\rangle_{U}} \qquad [27]$$

The free energy differences are not estimated directly. Instead, an averaging of the difference in potential energy between the mutant and the wild-type proteins is performed, which requires only one trajectory of the wild-type protein.[31,33,157]

ϕ-values can be estimated by Eq. [28] provided that entropic effects of the mutations are negligible and the internal energy of a protein is approximately proportional to the number of native contacts formed (as is the case for Gō models):[157]

$$\phi = \frac{\langle N\rangle_{\dagger} - \langle N\rangle_{U}}{\langle N\rangle_{N} - \langle N\rangle_{U}} \qquad [28]$$

The quantity $\langle N\rangle_L$ denotes the ensemble average number of native contacts in a macroscopic state L (with L corresponding to the unfolded U, folded N or transition † states). A simplification of Eq. [28] results when the fraction of native contacts in the unfolded state can be neglected:

$$\phi = \frac{\langle N\rangle_{\dagger}}{\langle N\rangle_{N}} \qquad [29]$$

The class of proteins for which Eq. [29] holds true are those possessing transition states that are very close, structurally, to the native state.[173] In addition to point mutations, computer simulations allow one to introduce perturbations by modifying/canceling native contacts. An analog of Eq. [29] for such perturbations is Eq. [30]:[32,172]

$$\phi = \frac{P_{ij}^{\dagger} - P_{ij}^{U}}{P_{ij}^{N} - P_{ij}^{U}} \qquad [30]$$

where P_{ij}^{N}, P_{ij}^{U} and $P_{ij}^{\dagger}$ denote the probability of formation for the "mutated" contact *ij* in the native (N), unfolded (U), and transition ($\dagger$) states, respectively. Figure 19 summarizes the variety of methods available for computing ϕ-values in computer simulations.

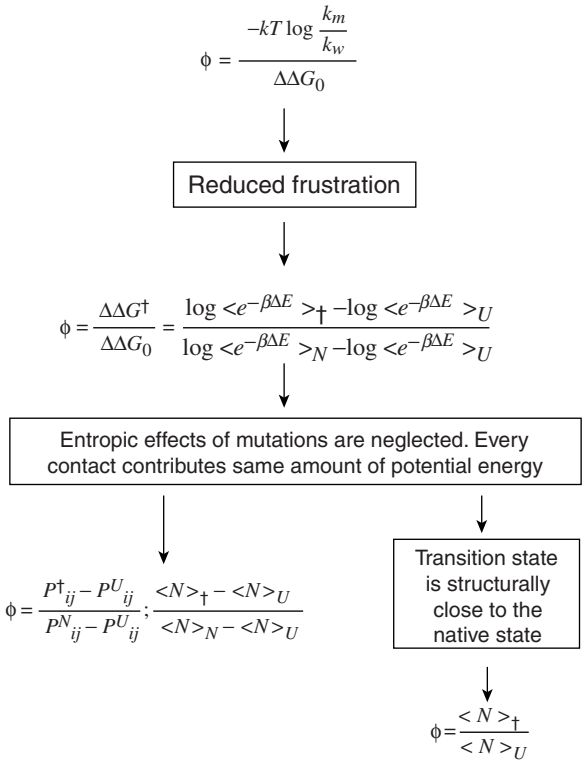


Figure 19 Numerical methods available for the evaluation of ϕ. The expressions rely on specific approximations, as discussed in detail in the main body of the text. Here ΔE is the energy difference between the mutated and the wild-type protein, P_{ij}^{L} is the probability of contact formation for the *ij* pair in state L, and $\langle N \rangle_L$ is the ensemble average number of native contacts in state L (unfolded, native, or transition). The remaining terms have been introduced in the previous figures.

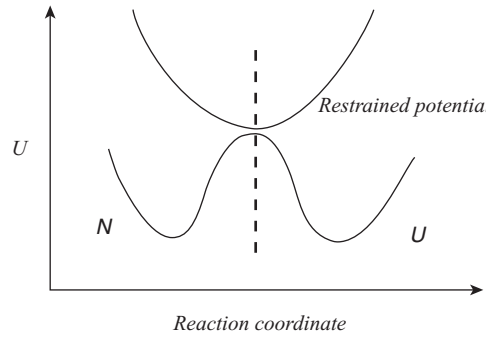


Figure 20 The method of Vendruscolo et al.[143] recognizes the fact that transition state structures are difficult to observe because they are located at the top of the folding free energy barrier and are thus rarely visited in simulations. To increase the statistical weight of these structures, a biasing potential is introduced that has a minimum in the TS region and drives the system away from both native and unfolded states. As a result, the free energy profile undergoes a transformation from a bimodal shape with populated native and denatured states to a unimodal shape where mostly TS structures are present.

Equation [29] serves as the basis for a numerical method of computing TSEs developed by Vendruscolo et al.[143] The method builds on the observation that transition state structures are intrinsically unstable; they are located at the top of the folding free energy barrier and are thus rarely visited in simulations. To enable a more ready determination of transition state conformations in simulations, Vendruscolo et al.[143] proposed to increase their statistical weight by imposing a biasing or restraining potential that forces sampling of the conformational space around the TS region. A straightforward way to accomplish this is to identify variables that describe the TSE uniquely and then to impose a potential that has a minimum around the TS values of those variables. Figure 20 illustrates the effect on the free energy profiles when such external potentials are introduced. Under these conditions, the free energy barrier is transformed into a free energy minimum.[174] Vendruscolo et al.[143] proposed that ϕ-values be used as the parameter that defines a transition state. Equation [29], introduced by Li and Daggett in their early evaluations of the TSE,[125] is employed to estimate ϕ-values from simulations. The quadratic form in ϕ that enforces sampling of the TSE is employed as the restraining potential:

$$E_{\min}(\Gamma) = \frac{\langle N \rangle_N}{N_\phi} \sum_i [\phi_i^{\exp} - \phi_i^{MD}(\Gamma)]^2 \qquad [31]$$

Here $\phi_i^{\exp}$ denotes the set of ϕ-values available from experiment, N_ϕ is the total number of those values, and the molecular dynamics $\phi_i^{MD}(\Gamma)$ is taken from Eq. [29] for a given conformation Γ. Even though this method enforces

sampling of conformations with ϕ-values close to those determined experimentally, Eq. [31] cannot reproduce most characteristics of the TSE structures reliably. For instance, the degree of compactness of the TSE, as measured by the radius of gyration, was found to be largely underestimated when simulations are run using Eq. [31] alone.[143] To better control sampling of TS structures, an additional potential term is thus introduced:

$$E_\lambda(\Gamma) = \lambda E_{\min}(\Gamma) + (1 - \lambda)E_{Go}(\Gamma) \quad [32]$$

$E_{Go}(\Gamma)$ is taken to be a Gō-type potential, which is designed to drive the protein toward its native state. The relative importance of the two terms $E_{\min}$ and E_{Go} is controlled by the adjustable parameter λ. This parameter is fine-tuned to ensure that exposure of the TS structures to the surrounding water molecules matches that measured in experiment. Using Monte Carlo simulations of a C_α-based model, it was found that $\lambda = 0.85$ can properly reproduce the experimentally observed degree of solvent exposure for the TSE of acylphos-phatase.[145]

By design, the restraining potential (Eq. [31]) ensures that all $\phi_i^{MD}(\Gamma)$ sampled in a simulation are distributed sharply around the experimental values $\phi_i^{\exp}$. It should be noted, however, that experiments provide information about average values of ϕ only and not about their distributions. A variety of different distributions in ϕ can produce the same average, simple examples of which include unimodal and bimodal distributions. It is unknown, a priori, what type of ϕ distributions are appropriate for a given protein; using Eq. [31] may thus introduce distortions into the computed TSE by enforcing a unimodal distribution of ϕ. This serious drawback of the original Vendruscolo method has been corrected.[144] The main modification concerns the way in which MD-derived ϕ-values are evaluated. Rather than computing ϕ_i^{MD} from Eq. [31] for a given conformation Γ, several identical yet independent copies (replicas) of the original system are run in parallel, and ϕ_i^{MD} is estimated as an *ensemble* average over these replicas. The advantage of this improved ensemble Monte Carlo method is that the type of ϕ-distribution is not assigned arbitrarily. Instead, the protein is free to pick the one that is dictated by its topology and specifics of its sequence. The most important outcome of this improvement is that the restraining potential does not limit the number of critical nuclei that comprise the transition state.[143]

An extension of the Vendruscolo method involves computer simulations and experimental measurements that are carried out in an iterative manner to make more efficient use of the information encoded in ϕ-values.[145] To map out the transition state, protein engineering is usually performed on all residues that are amenable to conservative mutations. The iterative method proposed by Paci et al. eliminates most of this labor-intensive and time-consuming process by reducing the number of mutations that need to be performed experimentally. The method is based on the premise that the critical nucleus

(CN), which represents the entire TSE as an "average" structure, is defined by only a few residues. Performing ensemble Monte Carlo simulations using the ϕ-values of the CN residues ensures that the ϕ-values of residues not belonging to the CN are still represented correctly. Hence, it is sufficient to determine the ϕ-values of the critical residues alone to reconstruct accurately the TSE. An iterative procedure is introduced to determine which residues comprise the CN. The procedure is initiated with a guess of the CN and the measurement of the ϕ-values for that presumed nucleus. In the first step of the iterative process (step A), computer simulations are run using the measured ϕ-values and the corresponding TSE is generated. ϕ-values are then computed (via simulation) for this putative ensemble and analyzed in terms of their variation. A residue belonging to the CN should have a simulated ϕ-value distributed narrowly around unity. Contrarily, residues not included in the restraining potential will show a wide distribution of ϕ-values. By monitoring the width of the computed ϕ-distribution, other residues can be identified that need to be included in the experiment, thus improving on the initial guess of the TSE determined. In this next step of the procedure (step B), residues displaying the greatest variation in ϕ are chosen for further experimental measurement. The newly obtained experimental ϕ-values are thus used as input for the Monte Carlo simulations, in which the procedure returns to step A with a larger set of residues. The iterations between steps A and B are continued until the convergence of ϕ-values at two consecutive iterative steps is observed, or until there are no remaining residues whose ϕ-values can be measured. For a fibronectin type III domain protein,[145] the method showed that ϕ-values for only 30 residues needed to be measured to identify the structure of the TSE successfully (only one third of the total number of residues of the protein). Significantly, only three theory/experiment iterations were needed to achieve convergence in the properties of the TSE, demonstrating the efficiency of this technique.

To conclude this section, we provide a short list of examples validating ϕ-value measurements as a useful tool for examining protein transition state structures. For each citation, we include the numerical method that uses these measurements to reconstruct transition state structures in computer simulations. Li and Daggett[125] found for the CI2 protein that the ϕ-values of its TS conformations obtained using the clustering analysis are in good agreement with those measured experimentally. Later work for the same protein showed[156,163] that the TSE conformations derived on the basis of experimental ϕ-values exhibit $P_{fold} \sim 1/2$. A similar result was obtained in the kinetics simulations of Gsponer and Caflisch[158] for the src SH3 domain. Their simulations were started from putative TS conformations constructed using the ϕ-value approach, and they found that trajectories had equal chances to fold or to unfold. Good agreement between computed and measured ϕ-values was also observed for the AcP protein.[143] It seems that the ϕ-value method of identifying the TSE is applicable to a wide class of proteins. We note a failure in the case of protein G,[166] however, where conformations predicted to

belong to the TSE had average P_{fold} close to the desired 1/2 but were distributed bimodally.

CONCLUSIONS AND FUTURE DIRECTIONS

We have presented an overview of simulation methodologies currently used to study the thermodynamics and kinetics of protein folding. Modern approaches range from very simplified lattice models, to more sophisticated off-lattice models, to atomically detailed, solvated systems. Taken together, the coarse-grained and detailed descriptions of proteins can provide a comprehensive picture of folding and augment and guide experimental studies.

This review has focused on the folding of small, two-state folders in idealized, dilute environments (in vitro folding). In everyday life, however, proteins fold in the more complex cellular environment and they often do not behave as simple two-state folders. Recent experimental advances[175,176] offer new insights into how in vivo environment affects folding, but we still lack a theoretical grasp of the complications that originate within the cell. Crowding in the cell, sequence mutations, and changes in pH or temperature can cause the folding process to go astray and lead to improperly folded entities. A particularly deleterious result is the self-assembly of misfolded structures into large, insoluble fibrillar aggregates. These aggregates are often lethal to the cell because of the resulting deficit of functional proteins or induced toxicity from both small soluble oligomers and larger fibrils.[175] Several seemingly unrelated diseases, including type II diabetes,[177] Creutzfeld-Jacob,[178] and Alzheimer's disease,[179] and certain forms of cancer, come from the misfolding of specific proteins. The cell has evolved several defense mechanisms to protect itself from protein misfolding, crowding, and aggregation. Chaperone-mediated folding is one of the most important, yet most poorly understood, of these mechanisms.[180] Chaperones are macromolecules, often proteins themselves, that recognize incorrectly folded proteins, assist folding, and in some cases can even reverse aggregation.

One theoretical challenge in the next decade will be to model protein folding in a cellular environment and develop a sound, theoretical framework for in vivo folding. Progress is already being made in this direction, with simulations on the effects of crowding,[181] the process of aggregation,[182–188] and the mechanism of chaperone-assisted folding[189–196] underway in several research groups.

ACKNOWLEDGMENTS

J.-E. Shea gratefully acknowledges financial support from the National Science Foundation (Career Award 0133504.), the David and Lucile Packard Foundation, and the A. P. Sloan Research Foundation. M. F. was supported by an NSF graduate research fellowship.

REFERENCES

1. C. Chothia, *Nature*, **357**, 543 (1992). One Thousand Families for the Molecular Biologist.
2. E. Haber and C. B. Anfinsen, *J. Biol. Chem.*, **237**, 1839 (1962). Side-chain Interactions Governing the Pairing of Half-cystine Residues in Ribonuclease.
3. C. B. Anfinsen, E. Haber, M. Sela, and F. H. White, Jr., *Proc. Natl. Acad. Sci. USA*, **47**, 1309 (1961). The Kinetics of Formation of Native Ribonuclease during Oxidation of the Reduced Polypeptide Chain.
4. C. B. Anfinsen, *Science*, **181**, 223 (1973). Principles that Govern the Folding of Protein Chains.
5. C. Levinthal, *J. Chimie Physique et de Phisio-Chemie Biologique*, **65**, 44 (1968). Are There Pathways for Protein Folding?
6. J. D. Bryngelson and P. G. Wolynes, *Proc. Natl. Acad. Sci. USA*, **84**, 7524 (1987). Spin Glasses and the Statistical Mechanics of Protein Folding.
7. J. D. Bryngelson and P. G. Wolynes, *J. Phys. Chem.*, **93**, 6902 (1989). Intermediates and Barrier Crossing in a Random Energy Model (with Applications to Protein Folding).
8. J. N. Onuchic, Z. Luthey-Schulten, and P. G. Wolynes, *Annu. Rev. Phys. Chem.*, **48**, 545 (1997). Theory of Protein Folding: The Energy Landscape Perspective.
9. K. A. Dill, S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas, and H. S. Chan, *Protein Sci.*, **4**, 561 (1995). Principles of Protein Folding–a Perspective from Simple Exact Models.
10. M. Karplus and D. L. Weaver, *Nature*, **260**, 404 (1976). Protein Folding Dynamics.
11. M. Karplus and D. L. Weaver, *Protein Sci.*, **3**, 650 (1994). Protein Folding Dynamics: Diffusion-collision Model and Experimental Data.
12. P. S. Kim and R. L. Baldwin, *Annu. Rev. Biochem.*, **51**, 459 (1982). Specific Intermediates in the Folding Reactions of Small Proteins and the Mechanism of Protein Folding.
13. W. Kauzmann, *Adv. Protein Chem.*, **14**, 1 (1959). Some Factors in the Interpretation of Protein Denaturation.
14. O. Ptitsyn, *Nature Struct. Biol.*, **3**, 488 (1996). How Molten is the Molten Globule?
15. D. B. Wetlaufer, *Proc. Natl. Acad. Sci. USA*, **70**, 697 (1973). Nucleation, Rapid Folding, and Globular Intrachain Regions in Proteins.
16. D. B. Wetlaufer, *Trends Biochem. Sci.*, **15**, 414 (1990). Nucleation in Protein Folding—Confusion of Structure and Process.
17. A. R. Fersht, *Curr. Opin. Struct. Biol.*, **7**, 3 (1997). Nucleation Mechanisms in Protein Folding.
18. J.-E. Shea and C. L. Brooks, III, *Annu. Rev. Phys. Chem.*, **52**, 499 (2001). From Folding Theories to Folding Proteins: A Review and Assessment of Simulation Studies of Protein Folding and Unfolding.
19. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. N. Teller, and E. Teller, *J. Chem. Phys.*, **21**, 1087 (1953). Equation of State Calculations by Fast Computing Machines.
20. K. Binder and A. P. Young, *Rev. Mod. Phys.*, **58**, 801 (1986). Spin Glasses: Experimental Facts, Theoretical Concepts and Open Questions.
21. E. Gardner and B. Derrida, *J. Stat. Phys.* **39**, 267 (1985). Zero Temperature Magnetization of a One-dimensional Spin Glass.
22. B. Derrida, *Physica D*, **107**, 186 (1997). From Random Walks to Spin Glasses.
23. P. W. Anderson, *J. Less-Common Metals*, **62**, 291 (1978). The Concept of Frustration in Spin Glasses.
24. M. Mezard, G. Parisi, and M. A. Virasoro, *Spin Glass Theory and Beyond*, World Scientific, Singapore, 1987.
25. J. D. Bryngelson, J. N. Onuchic, N. D. Socci, and P. G. Wolynes, *Proteins*, **21**, 167 (1995). Funnels, Pathways, and the Energy Landscape of Protein Folding: A Synthesis.

26. B. Derrida, *Phys. Rev. B*, **24**, 2613 (1981). Random-Energy Model: An Exactly Solvable Model of Disordered Systems.
27. B. Derrida, *Phys. Rev. Lett.*, **45**, 79 (1980). Random-Energy Model: Limit of a Family of Disordered Models.
28. T. Garel and H. Orland, *Europhys. Lett.*, **6**, 307 (1988). Mean-field Model for Protein Folding.
29. E. I. Shakhnovich and A. M. Gutin, *Biophys. Chem.*, **34**, 187 (1989). Formation of Unique Structure in Polypeptide Chains. Theoretical Investigation with the Aid of a Replica Approach.
30. P. E. Leopold, M. Montal, and J. N. Onuchic, *Proc. Natl. Acad. Sci. USA*, **89**, 8721 (1992). Protein Folding Funnels: A Kinetic Approach to the Sequence-Structure Relationship.
31. J.-E. Shea, J. N. Onuchic, and C. L. Brooks, III, *Proc. Natl. Acad. Sci. USA*, **96**, 12512 (1999). Exploring the Origins of Topological Frustration: Design of a Minimally Frustrated Model of Fragment B of Protein A.
32. J.-E. Shea, J. N. Onuchic, and C. L. Brooks, III, *J. Chem. Phys.*, **113**, 7663 (2000). Energetic Frustration and the Nature of the Transition State in Protein Folding.
33. C. Clementi, H. Nymeyer, and J. N. Onuchic, *J. Mol. Biol.*, **298**, 937 (2000). Topological and Energetic Factors: What Determines the Structural Details of the Transition State Ensemble and "en-route" Intermediates for Protein Folding? An Investigation for Small Globular Proteins.
34. E. Rhoades, E. Gussakovsky, and G. Haran, *Proc. Natl. Acad. Sci. USA*, **100**, 3197 (2003). Watching Proteins Fold one Molecule at a Time.
35. E. A. Lipman, B. Schuler, O. Bakajin, and W. A. Eaton, *Science*, **301**, 1233 (2003). Single-Molecule Measurement of Protein Folding Kinetics.
36. B. Schuler, E. A. Lipman, and W. A. Eaton, *Nature*, **419**, 743 (2002). Probing the Free-energy Surface for Protein Folding with Single-molecule Fluorescence Spectroscopy.
37. K. A. Dill, *Protein Sci.*, **8**, 1166 (1999). Polymer Principles and Protein Folding.
38. M. Gruebele, *Curr. Opin. Str. Biol.*, **12**, 161 (2002). Protein Folding: The Free Energy Surface.
39. C. M. Dobson and P. J. Hore, *Nature Struct. Biol.*, **5**, 504 (1998). Kinetic Studies of Protein Folding Using NMR Spectroscopy.
40. S. E. Jackson, *Folding Design*, **3**, R81 (1998). How do Small Single-domain Proteins Fold?
41. A. R. Fersht, *Structure and Mechanism in Protein Science*, W. H. Freeman and Company, New York, 1999.
42. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford, 1987.
43. B. A. Berg and T. Neuhaus, *Phys. Rev. Lett.*, **68**, 9 (1992). Multicanonical Ensemble: A New Approach to Simulate First-Order Phase Transition.
44. A. P. Lyubartsev, A. A. Martsinovski, S. V. Shevkunov, and P. N. Vorontsov-Velyaminov, *J. Chem. Phys.*, **96**, 1776 (1992). New Approach to Monte Carlo Calculation of the Free Energy: Method of Expanded Ensembles.
45. E. Marinari and G. Parisi, *Europhys. Lett.*, **19**, 451 (1992). Simulated Tempering: A New Monte Carlo Scheme.
46. B. Hesselbo and R. B. Stinchcombe, *Phys. Rev. Lett.*, **74**, 2151 (1995). Monte Carlo Simulation and Global Optimization without Parameters.
47. D. D. Frantz, D. L. Freeman, and J. D. Doll, *J. Chem. Phys.*, **74**, 2769 (1990). Reducing Quasi-ergodic Behavior in Monte Carlo Simulations by J-walking: Applications to Atomic Clusters.
48. C. Tsallis, *J. Stat. Phys.*, **52**, 479 (1988). Possible Generalization of Boltzmann-Gibbs Statistics.
49. U. H. E. Hansmann, *Physica A*, **242**, 250 (1997). Simulated Annealing with Tsallis Weights-A Numerical Comparison.

50. I. I. Andricioaei and J. E. Straub, *Phys. Rev. E.*, **53**, R3055 (1996). Generalized Simulated Annealing Algorithms using Tsallis Statistics: Application to Conformational Optimization of a Tetrapeptide.
51. F. Wang and D. P. Landau, *Phys. Rev. Lett.*, **86**, 2050 (2001). Efficient, Multiple-range Random Walk Algorithm to Calculate the Density of States.
52. R. H. Swendsen and J. S. Wang, *Phys. Rev. Lett.*, **57**, 2607 (1986). Replica Monte Carlo Simulation of Spin Glasses.
53. M. C. Tesi, E. J. J. van Rensberg, E. Orlandi, and S. G. Whittington, *J. Stat. Phys.*, **82**, 155 (1996). Monte Carlo Study of the Interacting Self-avoiding Walk Model in Three Dimensions.
54. K. Hukushima and K. Nemoto, *J. Phys. Soc. Jpn.*, **65**, 1604 (1996). Exchange Monte Carlo Method and Application to Spin Glass Simulations.
55. A. Mitsutake, Y. Sugita, and Y. Okamoto, *Biopolymers*, **60**, 96 (2001). Generalized-ensemble Algorithms for Molecular Simulations of Biopolymers.
56. A. M. Ferrenberg and R. H. Swendsen, *Phys. Rev. Lett.*, **63**, 1195 (1989). Optimized Monte Carlo Data Analysis.
57. A. M. Ferrenberg and R. H. Swendsen, *Phys. Rev. Lett.*, **61**, 2635 (1988). New Monte Carlo Technique for Studying Phase Transitions.
58. S. Kumar, D. Bouzida, R. H. Swendsen, P. A. Kollman, and J. M. Rosenberg, *J. Comput. Chem.*, **13**, 1011 (1992). The Weighted Histogram Analysis Method for Free-Energy Calculations on Biomolecules. I. The Method.
59. A. Šali, E. Shakhnovich, and M. Karplus, *Nature*, **369**, 248 (1994). How Does a Protein Fold?
60. N. D. Socci and J. N. Onuchic, *J. Chem. Phys.*, **101**, 1519 (1994). Folding Kinetics of Proteinlike Heteropolymers.
61. A. Kolinski, M. Milik, and J. Skolnick, *J. Chem. Phys.*, **94**, 3978 (1991). Static and Dynamic Properties of a New Lattice Model of Polypeptide Chains.
62. N. Gō and H. Taketomi, *Proc. Natl. Acad. Sci. USA*, **75**, 559 (1975). Respective Roles of Short and Long Range Interactions in Protein Folding.
63. H. Taketomi, Y. Ueda, and N. Gō, *Int. J. Pept. Protein Res.*, **7**, 445 (1975). Studies on Protein Folding, Unfolding and Fluctuations by Computer Simulation. I. The Effect of Specific Amino Acid Sequence Represented by Specific Inter-unit Interactions.
64. H. Nymeyer, A. E. Garcia, and J. N. Onuchic, *Proc. Natl. Acad. Sci. USA*, **95**, 5921 (1998). Folding Funnels and Frustration in Off-lattice Minimalist Protein Landscapes.
65. R. Du, V. S. Pande, A. Y. Grosberg, T. Tanaka, and E. I. Shakhnovich, *J. Chem. Phys.*, **108**, 334 (1998). On the Transition Coordinate for Protein Folding.
66. K. F. Lau and K. A. Dill, *Macromolecules*, **22**, 3986 (1989). A Lattice Statistical Mechanics Model of the Conformational and Sequence Spaces of Proteins.
67. H. S. Chan and K. A. Dill, *J. Chem. Phys.*, **92**, 3118 (1990). The Effects of Internal Constraints on the Configuration of Chain Molecules.
68. H. S. Chan and K. A. Dill, *Annu. Rev. Biophys. Biophys. Chem.*, **20**, 447 (1991). Polymer Principles in Protein Structure and Stability.
69. H. S. Chan and K. A. Dill, *J. Chem. Phys.*, **95**, 3775 (1991). "Sequence Space Soup" of Proteins and Copolymers.
70. H. S. Chan and K. A. Dill, *J. Chem. Phys.*, **99**, 2116 (1993). Energy Landscapes and the Collapse Dynamics of Homopolymers.
71. H. S. Chan and K. A. Dill, *J. Chem. Phys.*, **100**, 9238 (1994). Transition States and the Folding Dynamics of Proteins and Heteropolymers.
72. C. J. Camacho and D. Thirumalai, *Proc. Natl. Acad. Sci. USA*, **90**, 6369 (1993). Kinetics and Thermodynamics of Folding in Model Proteins.
73. E. I. Shakhnovich and A. M. Gutin, *Proc. Natl. Acad. Sci. USA*, **90**, 7195 (1993). Engineering of Stable and Fast-folding Sequences of Model Proteins.

74. A. Sali, E. Shakhnovich, and M. Karplus, *J. Mol. Biol.*, **235**, 1614 (1994). Kinetics of Protein Folding. A Lattice Model Study of the Requirements for Folding to the Native State.

75. N. D. Socci and J. N. Onuchic, *J. Chem. Phys.*, **103**, 4372 (1995). Kinetic and Thermodynamic Analysis of Proteinlike Heteropolymers: Monte Carlo Histogram Technique.

76. J. Skolnick and A. Kolinski, *J. Mol. Biol.*, **221**, 499 (1991). Dynamic Monte Carlo Simulations of a New Lattice Model of Globular Protein Folding, Structure and Dynamics.

77. A. Godzik, J. Skolnick, and A. Kolinski, *Proc. Natl. Acad. Sci. USA*, **89**, 2629 (1992). Simulations of the Folding Pathway of Triose Phosphate Isomerase-type Alpha/Beta Barrel Proteins.

78. A. Kolinski, L. Jaroszweski, P. Rotkiewicz, and J. Skolnick, *J. Phys. Chem. B*, **102**, 4628 (1998). An Efficient Monte Carlo Model of Protein Chains. Modeling the Short-Range Correlations Between Side Group Centers of Mass.

79. A. Kolinski, P. Rotkiewicz, B. Ilkowski, and J. Skolnick, *Proteins*, **37**, 592 (1999). A Method for the Improvement of Threading-based Protein Models.

80. A. Kolinski, B. Ilkowski, and J. Skolnick, *Biophys. J.*, **77**, 2942 (1999). Dynamics and Thermodynamics of Beta-hairpin Assembly: Insights from Various Simulation Techniques.

81. A. Kolinski and J. Skolnick, *Polymer*, **45**, 511 (2004). Reduced Models of Proteins and their Applications.

82. Y. Zhou and M. Karplus, *Proc. Natl. Acad. Sci. USA*, **94**, 14429 (1997). Folding Thermodynamics of a Model Three-helix Bundle Protein.

83. J. D. Honeycutt and D. Thirumalai, *Proc. Natl. Acad. Sci. USA*, **87**, 3526 (1990). Metastability of the Folded States of Globular Proteins.

84. Z. Guo, D. Thirumalai, and J. D. Honeycutt, *J. Chem. Phys.*, **97**, 525 (1992). Folding Kinetics of Proteins: A Model Study.

85. J. D. Honeycutt and D. Thirumalai, *Biopolymers*, **32**, 695 (1992). The Nature of Folded States of Globular Proteins.

86. Z. Guo and D. Thirumalai, *Biopolymers*, **36**, 83 (1994). Kinetics of Protein Folding: Nucleation Mechanism, Time Scales and Pathways.

87. Z. Guo and C. L. Brooks, III, *Biopolymers*, **42**, 745 (1997). Thermodynamics of Protein Folding: A Statistical Mechanical Study of a Small All-beta Protein.

88. J.-E. Shea, Y. D. Nochomovitz, Z. Guo, and C. L. Brooks, III, *J. Chem. Phys.*, **109**, 2895 (1998). Exploring the Space of Protein Folding Hamiltonians: The Balance of Forces in a Minimalist Beta-barrel Model.

89. S. Takada, Z. Luthey-Schulten, and P. G. Wolynes, *J. Chem. Phys.*, **110**, 11616 (1999). Folding Dynamics with Nonadditive Forces: A Simulation Study of a Designed Helical Protein and a Random Heteropolymer.

90. A. V. Smith and C. K. Hall, *J. Mol. Biol.*, **312**, 187 (2001). Protein Refolding versus Aggregation: Computer Simulations on an Intermediate-resolution Protein Model.

91. D. K. Klimov and D. Thirumalai, *Proc. Natl. Acad. Sci. USA*, **97**, 2544 (2000). Mechanisms and Kinetics of Beta-hairpin Formation.

92. F. Ding, N. V. Dokholyan, S. V. Buldyrev, H. E. Stanley, and E. I. Shakhnovich, *J. Mol. Biol.*, **324**, 851 (2002). Molecular Dynamics Simulation of the SH3 Domain Aggregation Suggests a Generic Amyloidogenesis Mechanism.

93. M. S. Cheung, J. M. Finke, B. Callahan, and J. N. Onuchic, *J. Phys. Chem.* **B**, **107**, 11193 (2003). Exploring the Interplay Between Topology and Secondary Structural Formation in the Protein Folding Problem.

94. J. Shimada and E. I. Shakhnovich, *Proc. Natl. Acad. Sci. USA*, **99**, 11175 (2002). The Ensemble Folding Kinetics of Protein G From an All-atom {Monte Carlo} Simulation.

95. C. Clementi, A. E. Garcia, and J. N. Onuchic, *J. Mol. Biol.*, **326**, 933 (2003). Interplay Among Tertiary Contacts, Secondary Structure Formation and Side-chain Packing in the Protein Folding Mechanism: All-atom Representation Study of Protein L.

96. M. S. Cheung, A. E. Garcia, and J. N. Onuchic, *Proc. Natl. Acad. Sci. USA*, **99**, 685 (2002). Protein Folding Mediated by Solvation: Water Expulsion and Formation of the Hydrophobic Core Occur After the Structural Collapse.

97. A. M. Fernandez-Escamilla, M. S. Cheung, M. C. Vega, M. Wilmanns, J. N. Onuchic, and L. Serrano, *Proc. Natl. Acad. Sci. USA*, **101**, 2834 (2004). Solvation in Protein Folding Analysis: Combination of Theoretical and Experimental Approaches.

98. J. Karanicolas and C. L. Brooks, III, *Proc. Natl. Acad. Sci. USA*, **100**, 3954 (2003). The Structural Basis for Biphasic Kinetics in the Folding of the WW Domain From a Formin-binding Protein: Lessons for Protein Design?

99. J. M. Sorenson and T. Head-Gordon, *Folding and Design*, **3**, 523 (1998). The Importance of Hydration for the Kinetics and Thermodynamics of Protein Folding: Simplified Lattice Models.

100. L. Verlet, *Phys. Rev.*, **159**, 98 (1967). Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules.

101. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, D. J. Swaminathan, and M. Karplus, *J. Comput. Chem.*, **4**, 187 (1982). CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations.

102. W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, *J. Am. Chem. Soc.*, **117**, 5179 (1995). A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules.

103. S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta, and P. Weiner, *J. Am. Chem. Soc.*, **106**, 765 (1984). A New Force Field for Molecular Mechanical Simulation of Nucleic Acids and Proteins.

104. W. L. Jorgensen and J. Tirado-Rives, *J. Am. Chem. Soc.*, **110**, 1657 (1988). The OPLS Potential Function for Proteins - Energy Minimizations for Crystals of Cyclic-peptides and Crambin.

105. W. R. P. Scott, P. H. Hunenberger, I. G. Tironi, A. E. Mark, S. R. Billeter, J. Fennen, A. E. Torda, T. Huber, P. Kruger, and W. F. van Gunsteren, *J. Phys. Chem. A*, **103**, 3596 (1999). The GROMOS Biomolecular Simulation Program Package.

106. D. Bashford and D. A. Case, *Ann. Rev. Phys. Chem.*, **51**, 129 (2000). Generalized Born Models of Macromolecular Solvation Effects.

107. M. S. Lee, F. R. J. Salsbury, and C. L. Brooks, III, *J. Chem. Phys.*, **116**, 10606 (2002). Novel Generalized Born Methods.

108. M. S. Lee, M. Feig, F. R. J. Salsbury, and C. L. Brooks, III, *J. Comput. Chem.*, **24**, 1348 (2003). New Analytical Approximation to the Standard Molecular Volume Definition and its Application to Generalized Born Calculations.

109. J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, *J. Comput. Phys.*, **23**, 327 (1977). Numerical-Integration of Cartesian Equations of Motion of a System with Constraints - Molecular Dynamics of N-alkanes.

110. L. Duan, L. Wang, and P. A. Kollman, *Proc. Natl. Acad. Sci. USA*, **95**, 9897 (1998). The Early Stage of Folding of Villin Headpiece Subdomain Observed in a 200-Nanosecond Fully Solvated Molecular Dynamic Simulation.

111. L. Duan and P. A. Kollman, *Science*, **282**, 740 (1998). Pathways to a Protein Folding Intermediate Observed in a 1-Microsecond Simulation in Aqueous Solution.

112. E. M. Boczko and C. L. Brooks, III, *Science*, **269**, 393 (1995). First Principles Calculation of the Folding Free Energy of a Three-helix Bundle Protein.

113. Z. Guo, E. M. Boczko, and C. L. Brooks, III, *Proc. Natl. Acad. Sci. USA*, **94**, 10161 (1997). Exploring the Folding Free Energy Surface of a Three-helix Bundle Protein.

114. D. O. V. Alonso and V. Daggett, *Proc. Natl. Acad. Sci. USA*, **97**, 133 (2000). Staphylococcal Protein A: Unfolding Pathways, Unfolded States, and Differences Between the B and E Domains.

115. A. E. Garcia and J. N. Onuchic, *Proc. Natl. Acad. Sci. USA*, **100**, 13898 (2003). Folding a Protein in a Computer: An Atomic Description of the Folding/Unfolding of Protein A.

116. A. Ghosh, R. Elber, and H. A. Scheraga, *Proc. Natl. Acad. Sci. USA*, **99**, 10394 (2002). An Atomically Detailed Study of the Folding Pathways of Protein A with the Stochastic Difference Equation.

117. S. Sato, T. L. Religa, V. Daggett, and A. R. Fersht, *Proc. Natl. Acad. Sci. USA*, **101**, 6952 (2004). Testing Protein-Folding Simulations by Experiment: B Domain of Protein A.

118. Y. Bai, A. Karimi, H. J. Dyson, and P. E. Wright, *Protein Sci.*, **6**, 1449 (1997). Absence of a Stable Intermediate on the Folding Pathway of Protein A.

119. S. Bottomley, A. Popplewell, M. Scawen, T. Wan, B. Sutton, and M. Gore, *Protein Eng.*, **7**, 1463 (1994). The Stability and Unfolding of an IgG Binding Protein Based Upon the B Domain of Protein A from Staphylococcus Aureus Probed by Tryptophan Substitution and Fluorescence Spectroscopy.

120. R. Elber, A. Ghosh, and A. E. Cardenas, *Acc. Chem. Res.*, **35**, 396 (2002). Long Time Dynamics of Complex Systems.

121. A. E. Cardenas and R. Elber, *Proteins Struct. Funct. Genet.*, **51**, 245 (2003). Kinetics of Cytochrome C Folding: Atomically Detailed Simulations.

122. G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, *Chem. Phys. Lett.*, **246**, 122 (1995). Pairwise Solute Descreening of Solute Charges from a Dielectric Medium.

123. A. R. Dinner and M. Karplus, *J. Mol. Biol.*, **292**, 403 (1999). Is Protein Unfolding the Reverse of Protein Folding? A Lattice Simulation Analysis.

124. J. Tsai, M. Levitt, and D. Baker, *J. Mol. Biol.*, **291**, 215 (1999). Hierarchy of Structure Loss in MD Simulations of src-SH3 Domain Unfolding.

125. A. Li and V. Daggett, *Proc. Natl. Acad. Sci. USA*, **91**, 10430 (1994). Characterization of the Transition State of Protein Unfolding by Use of Molecular Dynamics: Chymotrypsin Inhibitor 2.

126. A. Li and V. Daggett, *J. Mol. Biol.*, **257**, 412 (1996). Identification and Characterization of the Unfolding Transition State of Chymotrypsin Inhibitor 2 by Molecular Dynamics Simulations.

127. T. Lazaridis and M. Karplus, *Science*, **278**, 1928 (1997). New View of Protein Folding Reconciled with the Old Through Multiple Unfolding Simulations.

128. M. Levitt, M. Hirshberg, R. Sharon, and V. Daggett, *Comput. Phys. Commun.*, **91**, 215 (1995). Potential Energy Function and Parameters for Simulations of the Molecular Dynamics of Proteins and Nucleic Acids in Solution.

129. F. B. Sheinerman and C. L. Brooks, III, *J. Mol. Biol.*, **278**, 439 (1998). Calculations on Folding of Segment B1 of Streptococcal Protein G.

130. W. Guo, S. Lampoudi, and J.-E. Shea, *Biophys. J.*, **85**, 61 (2003). Posttransition State Desolvation of the Hydrophobic Core of the src-SH3 Protein Domain.

131. W. Guo, S. Lampoudi, and J.-E. Shea, *Proteins Struct. Funct. Genet.*, **55**, 395 (2004). Temperature Dependence of the Free Energy Landscape of the src-SH3 Protein Domain.

132. J.-E. Shea, J. N. Onuchic, and C. L. Brooks, III, *Proc. Natl. Acad. Sci. USA*, **99**, 16064 (2002). Probing the Folding Free Energy Landscape of the src-SH3 Protein Domain.

133. B. J. Berne and J. E. Straub, *Curr. Opin. Struct. Biol.*, **7**, 181 (1997). Novel Methods of Sampling Phase Space in the Simulation of Biological Systems.

134. Y. Sugita and Y. Okamoto, *Chem. Phys. Lett.*, **314**, 141 (1999). Replica-Exchange Molecular Dynamics Method for Protein Folding.

135. J. W. Pitera and W. Swope, *Proc. Natl. Acad. Sci. USA*, **100**, 7587 (2003). Understanding Folding and Design: Replica-exchange Simulations of "Trp-cage" Miniproteins.

136. A. E. Garcia and K. Y. Sanbonmatsu, *Proteins*, **42**, 345 (2001). Exploring the Energy Landscape of a Beta Hairpin in Explicit Solvent.

137. R. Zhou, *Proc. Natl. Acad. Sci. USA*, **100**, 13280 (2003). Trp-cage: Folding Free Energy Landscape in Explicit Water.

138. M. Oliveberg, *Curr. Opin. Struct. Biol.*, **11**, 94 (2001). Characterization of the Transition States for Protein Folding: Towards a New Level of Mechanistic Detail in Protein Engineering Analysis.

139. V. Daggett and A. Fersht, *Nat. Rev. Mol. Cell Biol.*, **4**, 497 (2003). The Present View of the Mechanism of Protein Folding.

140. G. Hummer, *J. Chem. Phys.*, **120**, 516 (2004). From Transition Paths to Transition States and Rate Coefficients.

141. Z. Guo and D. Thirumalai, *Folding Design*, **2**, 377 (1997). The Nucleation-Collapse Mechanism in Protein Folding: Evidence for the Non-uniqueness of the Folding Nucleus.

142. N. V. Dokholyan, S. V. Buldyrev, H. E. Stanley, and E. I. Shakhnovich, *J. Mol. Biol.*, **296**, 1183 (2000). Identifying the Protein Folding Nucleus Using Molecular Dynamics.

143. M. Vendruscolo, E. Paci, C. M. Dobson, and M. Karplus, *Nature*, **409**, 641 (2001). Three Key Residues from a Critical Contact Network in a Protein Folding Transition State.

144. R. Davis, C. M. Dobson, and M. Vendruscolo, *J. Chem. Phys.*, **117**, 9510 (2002). Determination of the Structures of Distinct Transition State Ensembles for a Beta-sheet Peptide with Parallel Folding Pathways.

145. E. Paci, J. Clarke, A. Steward, M. Vendruscolo, and M. Karplus, *Proc. Natl. Acad. Sci. USA*, **100**, 394 (2003). Self-consistent Determination of the Transition State for Protein Folding: Application to a Fibronectin Type III Domain.

146. P. Hänggi, P. Talkner, and M. Borkovec, *Rev. Mod. Phys.*, **62**, 251 (1990). Reaction-rate Theory: Fifty Years After Kramers.

147. A. Baumketner and Y. Hiwatari, *Phys. Rev. E*, **66**, 011905 (2002). Diffusive Dynamics of Protein Folding Studied by Molecular Dynamics Simulations of an Off-lattice Model.

148. P. G. Bolhuis, C. Dellago, and D. Chandler, *Proc. Natl. Acad. Sci. USA*, **97**, 5877 (2000). Reaction Coordinates of Biomolecular Isomerization.

149. N. D. Socci, J. N. Onuchic, and P. G. Wolynes, *J. Chem. Phys.*, **104**, 5860 (1996). Diffusive Dynamics of the Reaction Coordinate for Protein Folding Funnels.

150. J. Kubelka, J. Hofrichter, and W. A. Eaton, *Curr. Opin. Struct. Biol.*, **14**, 76 (2004). The Protein Folding 'Speed Limit'.

151. A. Baumketner, J.-E. Shea, and Y. Hiwatari, *J. Chem. Phys.*, **121**, 1114 (2004). Improved Theoretical Description of Protein Folding Kinetics from Rotations in the Phase Space of Relevant Order Parameters.

152. M. M. Klosek, B. J. Matkowsky, and Z. Schuss, *Ber. Bunsenges Phys. Chem.*, **95**, 331 (1991). The Kramers Problem in the Turnover Regime: The Role of the Stochastic Separatrix.

153. A. R. Dinner and M. Karplus, *J. Phys. Chem. B*, **103**, 7976 (1999). The Thermodynamics and Kinetics of Protein Folding: A Lattice Model Analysis of Multiple Pathways with Intermediates.

154. D. K. Klimov and D. Thirumalai, *Proteins Struct. Funct. Genet.*, **43**, 465 (2001). Multiple Protein Folding Nuclei and the Transition State Ensemble in Two-state Proteins.

155. J. Gsponer and A. Caflisch, *Proc. Natl. Acad. Sci. USA*, **99**, 6719 (2002). Molecular Dynamics Simulations of Protein Folding from the Transition State.

156. L. Li and E. I. Shakhnovich, *Proc. Natl. Acad. Sci. USA*, **98**, 13014 (2001). Constructing, Verifying, and Dissecting the Folding Transition State of Chymotrypsin Inhibitor 2 with All-atom Simulations.

157. F. Ding, N. V. Dokholyan, S. V. Buldyrev, H. E. Stanley, and E. I. Shakhnovich, *Biophys. J.*, **83**, 3525 (2002). Direct Molecular Dynamics Observation of Protein Folding Transition State Ensemble.

158. J. Gsponer and A. Caflisch, *J. Mol. Biol.*, **309**, 285 (2001). Role of Native Topology Investigated by Multiple Unfolding Simulations of Four SH3 Domains.

159. C. Clementi, P. A. Jennings, and J. N. Onuchic, *J. Mol. Biol.*, **311**, 879 (2001). Prediction of Folding Mechanism for Circular-permuted Proteins.

160. H. Nymeyer, N. D. Socci, and J. N. Onuchic, *Proc. Natl. Acad. Sci. USA*, **97**, 634 (2000). Landscape Approaches for Determining the Ensemble of Folding Transition States: Success and Failure Hinge on the Degree of Frustration.

161. N. V. Dokholyan, L. Li, F. Ding, and E. I. Shakhnovich, *Proc. Natl. Acad. Sci. USA*, **99**, 13014 (2002). Topological Determinants of Protein Folding.

162. V. Daggett, *Acc. Chem. Res.*, **35**, 422 (2002). Molecular Dynamics Simulations of the Protein Unfolding/Folding Reaction.

163. D. De Jong, R. Riley, D. O. V. Alonso, and V. Daggett, *J. Mol. Biol.*, **319**, 229 (2002). Probing the Energy Landscape of Protein Folding/Unfolding Transition States.

164. D. K. Klimov and D. Thirumalai, *J. Mol. Biol.*, **282**, 471 (1998). Lattice Models for Proteins Reveal Multiple Folding Nuclei for Nucleation-collapse Mechanism.

165. D. K. Klimov and D. Thirumalai, *Chem. Phys.*, **307**, 251 (2004). Progressing From Folding Trajectories to Transition State Ensemble in Proteins.

166. I. A. Hubner, M. Oliveberg, and E. I. Shakhnovich, *Proc. Natl. Acad. Sci. USA*, **101**, 8354 (2004). Simulation, Experiment, and Evolution: Understanding Nucleation in Protein S6 Folding.

167. D. K. Klimov and D. Thirumalai, *J. Mol. Biol.*, **315**, 721 (2002). Stiffness of the Distal Loop Restricts the Structural Heterogeneity of the Transition State Ensemble in SH3 Domains.

168. C. Dellago, P. G. Bolhuis, and D. Chandler, in *Advances in Chemical Physics*, I. Prigogine and S. A. Rice, Eds., Wiley, New York, 2002, pp. 1–78. Transition Path Sampling.

169. G. M. Torrie and J. P. Valleau, *J. Comput. Phys.*, **23**, 187 (1977). Nonphysical Sampling Distributions in Monte Carlo Free-energy Estimation: Umbrella Sampling.

170. P. G. Bolhuis, C. Dellago, P. L. Geissler, and D. Chandler, *J. Phys.: Condens. Matter*, **12**, A147 (2000). Transition Path Sampling: Throwing Ropes Over Mountains in the Dark.

171. A. R. Fersht, A. Matouschek, and L. Serrano, *J. Mol. Biol.*, **224**, 771 (1992). The Folding of an Enzyme. I. Theory of Protein Engineering Analysis of Stability and Pathway of Protein Folding.

172. J. N. Onuchic, H. Nymeyer, A. E. Garcia, J. Chahine, and N. D. Socci, *Adv. Prot. Chem.*, **53**, 87 (2000). The Energy Landscape Theory of Protein Folding: Insights into Folding Mechanisms and Scenarios.

173. E. Paci, M. Vendruscolo, and M. Karplus, *Proteins*, **47**, 379 (2002). Native and Non-native Interactions Along Protein Folding and Unfolding Pathways.

174. M. Vendruscolo and E. Paci, *Curr. Opin. Str. Biol.*, **13**, 82 (2003). Protein Folding: Bringing Theory and Experiment Closer Together.

175. C. M. Dobson, *Trends Biochem. Sci.*, **24**, 329 (1999). Protein Misfolding, Evolution and Disease.

176. R. Tycko, *Curr. Opin. Chem. Biol.*, **4**, 500 (2000). Solid-State NMR as a Probe of Amyloid Fibril Structure.

177. A. Clark, S. B. Charge, M. K. Badman, D. A. MacArthur, and E. J. de Koening, *Biochem. Soc. Trans.*, **24**, 594 (1996). Islet Amyloid Polypeptide: Actions and Role in the Pathogenesis of Diabetes.

178. S. B. Prusiner, *Science*, **252**, 245 (1991). Molecular Biology of Prion Disease.

179. D. M. Walsh, D. M. Hartley, Y. Kusumoto, Y. Fezoui, M. M. Condron, A. Lomakin, G. B. Bebdek, D. J. Selkoe, and D. B. Teplow, *J. Biol. Chem.*, **274**, 25945 (1999). Amyloid Beta-protein Fibrillogenesis: Structure and Biological Activity of Protofibrillar Intermediates.

180. F. U. Hartl and M. Hayer-Hartl, *Science*, **295**, 1852 (2002). Molecular Chaperones in the Cytosol: From Nascent Chain to Folded Protein.

181. M. Friedel, D. J. Sheeler, and J.-E. Shea, *J. Chem. Phys.*, **118**, 8106 (2003). Effects of Confinement and Crowding on the Thermodynamics and Kinetics of Folding of a Minimalist Beta-barrel Protein.

182. B. Ma and R. Nussinov, *Protein Sci.*, **11**, 2335 (2002). Molecular Dynamics Simulations of Alanine Rich Beta-sheet Oligomers: Insight into Amyloid Formation.

183. J. Gsponer, U. Haberthur, and A. Caflisch, *Proc. Natl. Acad. Sci. USA*, **100**, 5154 (2003). The Role of Side-chain Interactions in the Early Steps of Aggregation: Molecular Dynamics Simulations of an Amyloid-froming Peptide from the Yeast Prion Sup35.

184. M. Friedel and J.-E. Shea, *J. Chem. Phys.*, **120**, 5809 (2004). Self-assembly of Peptides into a Beta-barrel Motif.

185. H. D. Nguyen and C. K. Hall, *Proc. Natl. Acad. Sci. USA*, **101**, 16180 (2004). Molecular Dynamics Simulations of Spontaneous Fibril Formation by Random-Coil Peptides.

186. R. I. Dima and D. Thirumalai, *Protein Sci.*, **11**, 1036 (2002). Exploring Protein Aggregation and Self-propagation Using Lattice Models: Phase Diagram and Kinetics.

187. D. K. Klimov and D. Thirumalai, *Structure*, **11**, 295 (2003). Dissecting the Assembly of Aβ 16–22 Amyloid Peptides into Antiparallel β-sheets.

188. F. Massi and J. E. Straub, *Proteins*, **42**, 217 (2001). Energy Landscape Theory for Alzheimer's Amyloid β-peptide Fibril Elongation.

189. A. I. Jewett, A. Baumketner, and J.-E. Shea, *Proc. Natl. Acad. Sci. USA*, **101**, 13192 (2004). Accelerated Folding in the Weak Hydrophobic Environment of a Chaperonin Cavity: Creation of an Alternate Fast Folding Pathway.

190. A. Baumketner, A. I. Jewett, and J.-E. Shea, *J. Mol. Biol.*, **332**, 710 (2003). Effects of Confinement in Chaperonin Assisted Protein Folding: Rate Enhancement by Decreasing the Roughness of the Folding Energy Landscape.

191. K. Gulukota and P. G. Wolynes, *Proc. Natl. Acad. Sci. USA*, **91**, 9292 (1994). Statistical Mechanics of Kinetic Proofreading in Protein Folding in Vivo.

192. H. S. Chan and K. A. Dill, *Proteins Struct. Funct. Genet.*, **24**, 345 (1996). A Simple Model of Chaperonin-mediated Protein Folding.

193. C. D. Sfatos, A. M. Gutin, V. I. Abkevich, and E. Shakhnovich, *Biochemistry*, **35**, 334 (1996). Simulations of Chaperone-assisted Folding.

194. M. Betancourt and D. Thirumalai, *J. Mol. Biol.*, **287**, 627 (1999). Exploring the Kinetic Requirements for Enhancement of Protein Folding Rates in the GroEL Cavity.

195. D. Gorse, *Biopolymers*, **59**, 411 (2001). Global Minimization of an Off-lattice Potential Energy Function Using a Chaperone-based Refolding Method.

196. D. Gorse, *Biopolymers*, **64**, 146 (2002). Application of a Chaperone-based Refolding Method to Two-and Three-dimensional Off-lattice Protein Models.

CHAPTER 4

The Simulation of Ionic Charge Transport in Biological Ion Channels: An Introduction to Numerical Methods

Marco Saraniti,[*] Shela Aboud,[†] and Robert Eisenberg[†]

[*]*Electrical and Computer Engineering Department, Illinois Institute of Technology, Chicago, IL*
[†]*Department of Molecular Biophysics and Physiology, Rush University, Chicago, IL*

INTRODUCTION

Ion channels are proteins embedded in the lipid membrane of biological cells. They interact in a complex way with their environment and are responsible for finely regulating the flux of ionic charge across the membrane. For instance, the generation and transmission of potentials in nerves and muscles, as well as the hormone release from endocrine cells, are believed to be mechanisms governed by the transport of ionic charge through these protein "gates."[1]

Since the demonstration in 1976[2] of a reliable experimental methodology for the detection of currents flowing through individual ion channels, several refinements of the experimental setup have been successfully applied to a variety of membrane and cell configurations, both in vivo and in vitro.[3] The extraordinary progress of those experimental techniques triggered an increasing theoretical effort aimed at the understanding of the role of ion channels in the physiology of complex biological systems, and, more generally, their influence on the electrical equilibrium between the cells and their environment. Besides the purely theoretical aspect, important pharmacological advances

Reviews in Computational Chemistry, Volume 22
edited by Kenny B. Lipkowitz, Thomas R. Cundari, and Valerie J. Gillet

have arisen from improved knowledge of ion channels.[4] Furthermore, from an engineering viewpoint, ion channels are being envisioned as a key component in a new generation of biosensors that integrate the selectivity and extreme sensitivity of ion channels with the processing capabilities of modern microelectronics.[5–7]

The appeal of the many possible applications of ion channels is only one part of the complete story. The interest of the computer modeling community has also been triggered by several concomitant events: (1) the availability of reliable protein structural data, (2) the capability of producing mutants by reprogramming the genetic sequence of bacteria, (3) the availability of reproducible experimental data on the electrophysiology of individual channels, and (4) the availability of adequate computational machinery (hardware *and* software) for the realistic modeling and simulation of ion channels in their environment. All these contributions occurred more or less simultaneously during the last decade, and produced a rather unusual synergy among experimentalists, theoreticians, and computational scientists who combined their efforts in order to relate the structure of ion channels to their function.

A peculiar aspect of the research on ion channels is that it frequently involves researchers working in traditionally different disciplines. The solid-state electronics community, for example, is well aware of the fact that traditional scaling – i.e., the reduction of the feature-size of transistors needed to increase the performance of integrated circuits[8] – will soon be inadequate to satisfy the requirements of emerging technologies.[9] A natural solution is to increase the complexity rather than the speed of the basic components, and much can be learned from ion channels, which are extremely specialized and miniaturized low power devices. Transistors are definitely faster than ion channels, but the advantage due to their operational speed is compensated by the complexity of the operations performed by ion channels. It appears clear that the full understanding of ion channel properties will allow for either the modification of their design for novel applications, or for manufacturing analogous structures capable of emulating their functionality.

This chapter is intended to be an introduction to the numerical techniques used for the simulation of charge transport through ion channels. The complexity and the size of the systems to be simulated will be stressed throughout the entire chapter, as well as the potential for the practical applications of ion channels in several fields. We firmly believe that the computer simulation of ion channels is not just a ''large scale'' problem that will be progressively solved as computer performance naturally evolves. High performance computing is only one component of the solution, and much work is needed for devising and integrating adequate physical models and algorithmic approaches. Therefore, ion channel simulation is a good example of the assertion that ''computers do not solve problems, people do.''[10]

In addition to the introduction, this chapter consists of four main sections and some concluding remarks. First, a description of the computational

methods used to model the electrostatic framework of ion channels is given, including a discussion of the boundary conditions traditionally used. A rather detailed description of efficient algorithms for the solution of Poisson's equation in real space is supplied for representing the long-range electrostatics of ion channel systems. This approach has been identified as a possible improvement of the force-field models used in particle-based simulation (molecular and Brownian dynamics),[11,12] but its popularity is low because it has been limited to simple test problems. The two following sections are devoted to the various models of ionic charge transport through the channels. In particular, a classification of continuum and particle-based methods is provided, and a discussion of their modeling capabilities is presented. The need for a hierarchy of numerical approaches needed to model the behavior of the systems of interest at different time and space scales is discussed in the penultimate section. The last section is devoted to problems that are still open, and to the future direction of research on the numerical simulation of biological ion channels. For reasons of space, this chapter focuses mainly on the numerical methods used to directly model charge transport in biological ion channels. However, it should be noted that a great deal of information on these systems and their properties is obtained with other techniques such as quantum chemistry (or structural ab initio) methods and by stochastic sampling approaches for the analysis of trajectories in the phase-space (Monte Carlo methods).

The remaining part of the introduction will be devoted to the description of the simulative environment required to model the operation of ion channels.

System Components

Ion channels interact strongly with their environment. From a microscopic viewpoint, these proteins cross the lipid bilayer that forms the cell membrane, and are exposed to the electrochemically different environments found inside and outside the cell. They are designed to react in a highly specialized way to specific stimuli – mechanical, chemical, or electrical – and to express their function by regulating the ionic flux across the cell membrane. For this reason, any simulative approach meant to model ion channels must account *in some way* for the combined behavior of the protein channel, the membrane, and the aqueous solution containing the ionic species of interest. Additionally, a way to represent a specific stimulus must be devised, in order to model the transient behavior of the channels as a function of the "external" perturbation.

The Protein

Because of the highly specialized functions they perform, ion channels are classified into different families. These families are based upon the ions those channels selectively allow to flow into and out of the cell. This functional classification[1] has been adopted as a result of the early electrophysiologic experiments on ion channels, and its success is due to the strict relation between the

structure of the channels and their function. It should be noted that several channels that allow the diffusive flow of non-ionized substances across the membrane exist, an example of which is the mechano-sensitive channels regulating the bacterial cytoplasmic pressure by responding to membrane tension.[13] Most of the basic models discussed in this chapter apply to these diffusive channels as well.

Most of the membrane proteins contain α-helices and β-sheets connected in structures of varying complexity. Within a specific structure, some helices cross the membrane from one side to the other, while other segments of the protein are confined in a more limited region. The amino-acid sequences are structured in such a way that one or more pores are formed inside the protein, that are large enough to allow for ionic flow. The protein structure is flexible and, in many cases, the functionality of ion channels is achieved by structural changes occurring in specific locations of the amino acid sequence.

From the functional viewpoint, we will discuss mainly the operations of ion channels in relation to the following three properties: *permeation*, which is the property of allowing ions to cross the strong dielectric barrier due to the cell membrane; *selectivity*, which is the capability of discriminating between the ionic species flowing through the channel structure; and *gating*, which is the capability of modulating the flux through the channel in response to an external stimulus. It is important to note that gating occurs on a time scale several orders of magnitude longer than the typical transit time of one ion though the channel pore. This is a critical aspect of these systems that must be accounted for in the simulation: The crucial physics occurs on distances measured in a few angstroms and starts on a femtosecond time scale, while the resulting physiological functionality is expressed in milliseconds and on distances measured in microns. The ability to relate the ultrafast microscopic processes occurring in channel proteins to their slow physiological expression is *the* challenge of ion channels simulation.

The structural features of some channels will be presented briefly in the remaining part of this introduction. This discussion is not meant to offer a classification of ion channels, but rather some key features of notable structures that are used as examples in this chapter. The three channels we now describe are: Gramicidin A, potassium channels and finally porins.

Gramicidin A. Gramicidin A (gA) is a small 15-residue antibiotic peptide formed as a dimer in a head-to-head (HH) or a double-helical (DH) conformation.[14] Because of its simplicity and reduced dimensions, the gA structure has been studied extensively and simulated as a model for ion channels,[15–19] and has emerged as a benchmark for simulation approaches.[20–24] The structure exposes its hydrophobic sidechains to the lipid membrane that embeds the protein. The molecular structure of gA has been known for three decades,[25] and has been recently resolved with NMR spectroscopy.[26,27] The relation of the structure seen spectroscopically to that

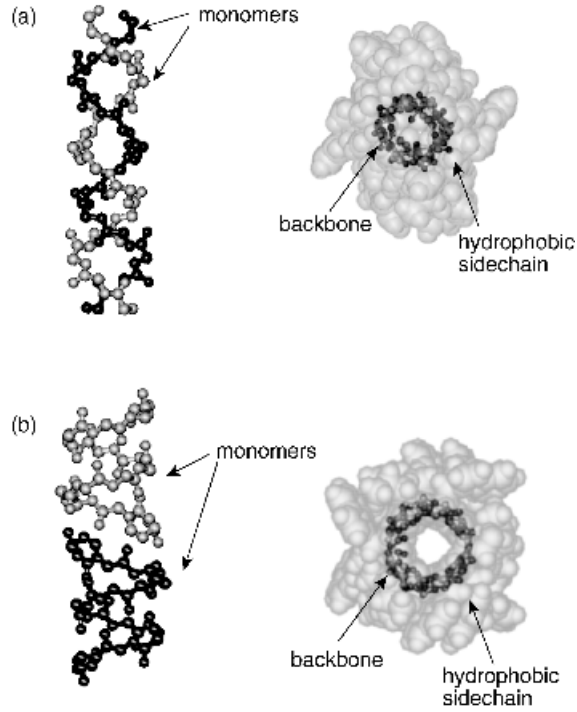


Figure 1 Atomic structure of a (a) double-helical (1mic.pdb[29]) and (b) head-to-head (1mag.pdb[30]) conformation of Gramicidin A. The pictures to the left show the backbone representation perpendicular to the pore. The pictures to the right show a view parallel to the channel that includes the hydrophobic sidechains. The pictures were generated with VMD.[39]

in membranes which conducts ions is being investigated.[28] Figure 1 shows the gA backbone structure perpendicular to the pore and a top view along the backbone structure showing the hydrophobic sidechains for the (a) DH[29] and (b) HH[30] conformations. The interaction of the protein with the lipid bilayer (hydrophobic matching) has been modeled quantitatively[31,32] and measured experimentally,[33] as has its properties of water transport.[14,34] Concerning gating, characteristically fast (sub-millisecond) closure events, called flickers, have been attributed to either conformational changes (lateral shifts of the monomers[35]) triggered by the breaking of the hydrogen bonds joining the dimer in the HH configuration, or by undulations of the bilayer that modify the conductive state of the protein.[36] Novel experimental techniques, such as patch-clamp fluorescence microscopy,[37] are being devised for detailed observation of the conformational changes of this simple structure. From a charge transport viewpoint, gA selects monovalent cations and its conductivity depends on both the membrane conformation and the ionic concentration surrounding it.[38]

Potassium Channels. Potassium channels, or K-channels are present in nearly all cells,[1] and play a key role in stabilizing the membrane potential in excitable cells. They are therefore crucial for the electrical functionality of the nervous system. They are characterized by an extreme selectivity (their permeability for K^+ is thousands of times larger than that of the smaller Na^+ ions), and by a high diffusivity (comparable to bulk water). The molecular structure of several K-channels has been disclosed by means of X-ray spectroscopy. In particular, a 3.2 Å resolution mapping of the pH-dependent bacterial KcsA channel was performed by Doyle et al.,[40] and a higher resolution (2.0 Å) structure has been subsequently disclosed by the same group.[41] The structure of the ligand-gated MthK channel, which opens in response to intracellular Ca^{2+}[42] has also been determined, and the structure of the voltage-gated[43] KvAP channel has been published recently.[44] Together with accurate structural data, several hypotheses about the functionality of these ion channels have been formulated.

The three K-channels listed above are tetrameric assemblies with sub-units sharing the same signature amino acid sequence TXGYGD of the selectivity filter. Also common is the topology of the ionic permeation channel, composed of two transmembrane helices (inner and outer helix) per sub-unit. The two transmembrane helices are joined by a pore sequence constructed with a shorter pore helix plus the selectivity filter segment.

The KcsA channel is characterized by this simple transmembrane architecture (see Ref. 40, and the left side of Fig. 2), and its activation has been attributed to pH-dependent translations and rotations of the two transmembrane helices.[45] Because of its relative simplicity, KcsA has been simulated extensively with a variety of approaches, and, like gA, it too can be considered a benchmark system for simulation codes.[38]

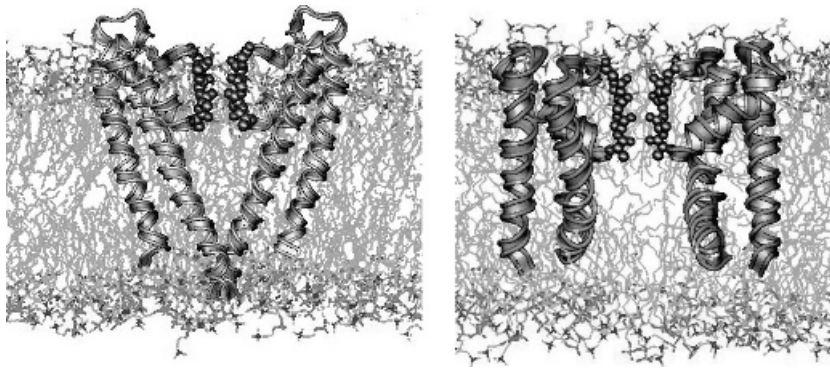


Figure 2 Two sub-units of the KcsA (left) and MthK (right) potassium channels embedded in an explicit POPC lipid bilayer. The atoms lining the selectivity filter are represented as spheres to show the individual "cages" which represent the binding sites of K^+ ions. The KcsA and MthK structures are obtained from the protein database codes 1bl8.pdb[40] and 1lnq.pdb,[42] respectively.

The MthK channel has an additional large gating ring below the membrane-spanning pore (right side of Fig. 2, the gating ring is not included). The gating ring is responsible for converting the free energy of intracellular Ca^{2+} into mechanical work that pulls apart the helices of the transmembrane pore and opens it to allow potassium permeation.[42] Finally, the sequence of the voltage-dependent KvAP channel shows six transmembrane helices: the same two hydrophobic segments of KcsA and MthK (segments S5 and S6) and four additional helices (S1–S4) that constitute the voltage-activated gate. A section of S3 (S3b) and S4 define a mobile "voltage-sensor paddle."[44] Recent experiments[46] suggested that, due to the presence of charged amino acids in S4, the channel undergoes a dramatic conformational change in the presence of an adequate transmembrane voltage: the "paddle" rotates more or less rigidly and crosses most of the lipid membrane pulling open the pore made by the helices S5 and S6. This interpretation of the gating mechanism of MthK is currently under intense investigation (see, for example, Ref. 47).

The permeation path of K-channels shows a very irregular (but highly functional) profile:[38] a hydrated ion moves (outward) initially through the intracellular gate made by the tips of the inner helices, enters a large central cavity (that probably favors monovalent cations over intracellular polyvalent cations[48]) filled with tens of water molecules, and then crosses the extremely narrow (angstrom size) selectivity filter where its solvation is at least partially due to carbonyl oxygen atoms rather than water.[41] A potassium ion therefore changes its hydration configuration during the journey, and travels an electrostatically irregular pathway to exit the cell. K-channels are engineered in such a way that this process is extremely fast and highly K-selective.

Porins. Porins are the first channels for which an atomic crystal structure was available.[1] These proteins function as ion channels with high conductivity and relatively low selectivity.[38] Because of the availability of experimental data,[49] porins have been used extensively to build and test simulation methods for ion channels. Many mutants of the bacterial trimer OmpF have been synthesized and modeled.[50] The permeation process has been simulated with different molecular dynamics approaches.[51,52] OmpF is a relatively large polypeptide made of three monomers composed of 340 amino acid each. The monomer is a hollow β-barrel structure formed by 16 antiparallel β-strands. The structure has eight loops (L1-L8) that form the water-filled pore. Loop L3 folds inside the barrel and generates a structural constriction that reduces the lumen of the pore to a diameter of approximately 6 Å. A top view of the three monomers of OmpF (protein database code 2omf.pdb[53]) is shown in Fig. 3 (right), where the L3 loops are represented by the large shaded cylinders. The OmpF crystal structure embedded in an explicit lipid membrane is also shown on the left side of Fig. 3. The charge distribution in the proximity of the constriction and all over the length of the pore, plays a crucial role in the permeation properties of OmpF. Furthermore, the

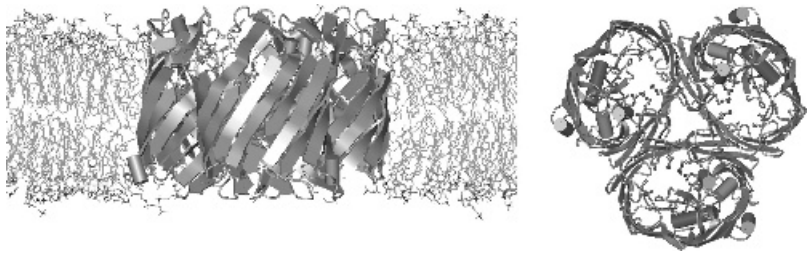


Figure 3 The OmpF porin channel (left) embedded in an explicit POPC membrane, and (right) the corresponding top view of the OmpF. The L3 loop in the constriction zone of the three monomers is represented by the large shaded cylinder. The structure has been obtained from the protein database (2ompf.pdb)[53] and the plot has been rendered with VMD.[39]

close proximity of negative and positive charges within the constriction zone generates an intense electric field that interacts with ions and determines the channel conductivity. The ionization state of the residues in the pore changes with the pH of the solution, suggesting that OmpF may function as a pH-gated channel in some conditions. The role of the conformational changes due to molecular flexibility (particularly for the L3 loop) is still an open question for the understanding of the functionality of OmpF.

The electrical properties of OmpF have been measured for long times, both with patch-clamp techniques and on planar lipid membranes.[49] A high-resolution electrostatic mapping of the trimer was obtained with atomic probe microscopy,[54] while a systematic electrostatic modeling of the pore lumen has been recently performed by two groups[50,51] who did not limit their study to the wild protein, but comparatively analyzed several mutants. The electrostatic landscape of OmpF is a typical example of how the balance between strong interactions finely tunes the properties of a channel. Ionic trajectories have been simulated both with Brownian and molecular dynamics simulation codes,[52] and the role of ion-ion interaction within the pore has been stressed as being important.

The Membrane

The cell membrane is made of amphipathic molecules consisting of one polar, hydrophilic head and one (or two) nonpolar, hydrophobic tails.[55] In an aqueous environment the lipid molecules spontaneously aggregate into conformations that minimize the interaction between water molecules and the hydrophobic tails of the lipids. One configuration that is energetically favorable is that where the lipid bilayer,[56] composed of two parallel sheets of lipid molecules, is oriented in such a way that the molecular heads are in contact with the aqueous solution and the tails are inside the membrane thickness. Under conditions of normal cell function, the lipid is an extremely stable two-dimensional structure that rapidly reassembles itself if disturbed or broken.

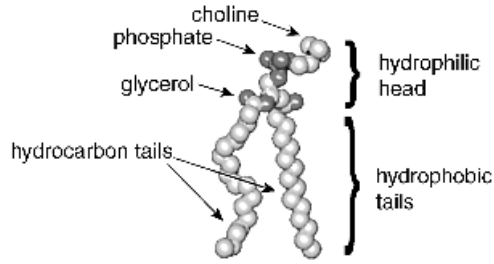


Figure 4 The head group of the lipid molecule phosphatidylcholine is composed by a choline, a phosphate and a glycerol, while the hydrophobic tails are formed from two fatty acid chains. The atomic coordinates of the lipid molecule are from the work of Tieleman *et al.*,[58] and the plot is rendered with VMD.[39]

The three main classes of lipids present in cell membranes are phospholipids, glycolipids and cholesterol.[57] Phospholipids are the most abundant of the biological membrane lipids and are assembled from fatty acids, alcohol and phosphate. The hydrophilic head is composed of an alcohol (such as choline) joined through a phosphate to either glycerol or sphingosine. Fatty acid hydrocarbon chains are attached to the lipid molecule through the glycerol or sphingosine and constitute the hydrophobic tails. The phospholipids based on glycerol are called phosphoglycerides while those based on sphingosine are called sphingolipids. The phosphatidylcholine (POPC) molecule, shown in Fig. 4 is the most common phosphoglyceride in biological cells,[57] and is characterized by a choline molecule attached to the phosphate at the hydrophilic head. Additionally, one of the hydrocarbon tails is fully saturated while the other contains several unsaturated bonds, creating the tail kinks shown in Fig. 4. The fluidity, or lateral diffusion of lipid molecules within the bilayer, depends on the length and saturation of the hydrocarbon tails. A cross section of a lipid bilayer formed with POPC molecules is shown in Fig. 5. Long hydrocarbon chains increase the "drag" on a lipid while unsaturated bonds improve the lipid mobility due to the reduction of the overall packing density. The phospholipid sphingomyelin is distinguished from POPC by a long hydrocarbon chain of sphingosine which substitutes for one of the fatty acids in the hydrophobic tails.

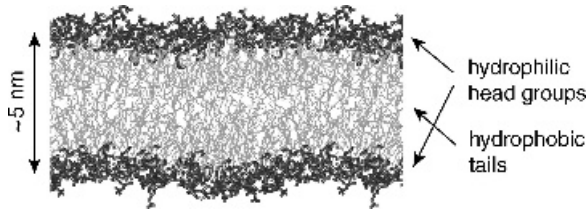


Figure 5 Cross section of a membrane composed of phosphatydylcholine molecules. The graphic rendering has been obtained with VMD.[39]

The second class of lipids are glycolipids. They are structurally similar to sphingomyelin except they contain sugar residues, such as glucose or galactose, instead of the phosphate-alcohol group in the hydrophilic head. The sugar residues in glycolipids are always oriented on the extracellular side of the membrane and form part of the carbohydrate coating that surrounds most animal cells.

Cholesterol is a steroid that has a different structure than either phospholipids or glycolipids. The body is constructed primarily of four hydrocarbon rings. The polar head is formed by a hydroxyl group attached at one end while a long saturated hydorphobic hydrocarbon tail is attached to the other end of the ring system. The steroid rings form a rigid planar structure that reduces the fluidity of the plasma membrane. In animal cells, cholesterol molecules are located between the phospholipids, filling the spaces from the kinked unsaturated bonds of the hydrocarbon tails thus making the lipid more rigid.

The precise composition of the lipid membrane in biological cells is inherently complex, and varies depending on both the species and type of cell. In addition, the local distribution of the lipid molecules within a single bilayer can be highly disordered, and, the two corresponding monolayers are generally asymmetric.[55] The inclusion of transmembrane structures, such as ion channel proteins and polymers, further complicates the picture. The extremely heterogeneous nature of the bilayer combined with the flexibility and polarizability of the lipid molecules makes the study of membranes in real biological systems a formidable task, both from the experimental and the computational viewpoint.

Because of the fluctuations of the flexible biological membranes, the structural characterization of lipid bilayers is an arduous task when atomic details are sought.[59] Indeed, structural information about the membrane thicknesses, such as the hydrophobic thickness and head group separation, as well as the lipid density, are very difficult to quantify. This results in a large uncertainty in the experimentally determined structural parameters of lipid bilayers found in the literature. For example, values of the average area per phospholipid molecule measured in a single lipid system can vary by nearly 30 $\mathring{A}^2$.[59]

The simulation of lipid bilayers provides a method for probing microscopic details of the lipid system, and relates those details to the macroscopic behavior observed experimentally.[56,60] The molecular dynamics approach is the most popular choice for membrane simulation, because it provides information about the spatial and temporal evolution of both single species phospholipid membranes,[61,62] and multi-lipid systems.[63–65] For example, molecular dynamics allowed for the characterization of phospholipid bilayers in terms of their interaction with water, and revealed that the orientation of the water molecules compensated for the fluctuations in the lipid head group, resulting in an almost constant membrane dipole potential.[61,66]

Although molecular dynamics is arguably the most accurate simulation technique, the characteristic relaxation times of the lipid system are generally

orders of magnitude larger than the time that is needed to obtain statistically significant results.[62,67] Also, the space scale over which the lipids self-organize can exceed the size of the computational domain that can be realistically simulated with extant molecular dynamics techniques.

The Aqueous Environment

Aqueous solutions under biologically relevant conditions are composed primarily of water molecules; water therefore plays a primary role in many chemical and physical processes.[68] Water is a highly polar molecule due to its bent configuration. A spatial separation exists between the internal positive and negative charges in the electrically neutral molecule, giving rise to a strong, permanent electric polarization field.[69] Furthermore, the separation between internal charges makes it possible for the oxygen atom of one water molecule to bond electrostatically to the hydrogen atoms in neighboring molecules. This hydrogen bonding facilitates the formation of relatively large domains of water molecules into lattice-type structures[69] analogous to crystalline ice. This cluster configuration of liquid water is not static and domains are continually formed and disassociated.

Ions in aqueous solution alter the structure of water in such a way that the water molecules will orient themselves around the charged ions with the appropriate polar side of the water pointing toward the ion, and creating one or more hydration shells. The water in the hydration shell now behaves differently than the bulk water in the sense that its dynamics is correlated with the ionic motion.

Aqueous solutions confined in regions of molecular dimensions, such as the narrow pores of ion channels, exhibit different properties than bulk ionic solutions, and one way to characterize the microscopic properties of ion channels is to identify these differences.[38] Confinement in small regions restricts the translational and rotational motion of the water molecules, and creates a greater degree of order. Simulations revealed a consequently strong decrease of the diffusion coefficient in small water-filled cavities. In addition to the effects of the physical confinement, a significant electrostatic interaction is also present between the water molecules and the cavity walls. In the case of porin channels, for example, the internal transverse electric field is so high that the cavity region is no longer a linear dielectric medium.[38] Polar groups in the pore lining interact with the water molecules in the hydration shell of ions as they traverse the pore, and this interaction may play a direct role in the selectivity properties of protein channels.[38,70]

Although not rigorously correct, the approximation of water as a structureless homogeneous continuum dielectric medium is used by many simulative methodologies. Both Brownian dynamics (see the section entitled Implicit Solvation: Brownian Dynamics) and electrodiffusive approaches (see the section on Flux-Based Simulation) include the water in the electrostatic picture as a continuous dielectric background with polarizability appropriately tuned

inside the channel pore. These "implicit water" models are able to reproduce activity coefficients for a variety of bulk systems[71–73] as well as the conductivity behavior of ions through channels.[22] Obviously, care must be taken when applying these techniques to model ion transport in channels where individual water molecules play a crucial role,[23] such as within the extremely narrow selectivity filter of potassium channels.

Representing the Full System

The definition of the system to be simulated, as well as the choice of the details in its representation, are crucial for the simulation of ionic transport in protein channel systems. Several components, including the channel itself, can be represented *implicitly*, that is through some macroscopic properties representative of their effects on the simulative landscape, or *explicitly*, that is with a microscopic, atomic-scale, resolution that is governed by fundamental laws. The model used for the computational representation depends on the specific questions to be addressed.[74]

For instance, an implicit membrane model greatly reduces the computational burden, and is appropriate in many cases because the lipid-protein interaction is often only important for protein stability and insertion.[75] In addition, the time scale of the charge transport process across the membrane is usually much longer than the time scale of the lipid fluctuations, so the motion of the charge is influenced primarily by the membrane through its dielectric rather than dynamics properties. Within the implicit bilayer representation, the membrane is treated as an impermeable slab of either a homogeneous dielectric material, or as a slab with a low dielectric constant in the region of the nonpolar tails and a higher effective dielectric constant in the region where the charged head groups of the lipids reside.[53,76]

When the interaction between membrane and protein channels becomes significant at the atomic level, an explicit representation of the molecules forming the lipid bilayer and protein channels must be built and modeled in such a way that the mechanical, chemical, and electrostatic properties of the system are modeled with appropriate detail. Two basics techniques generally have been used to build the computer representation of a channel/membrane system with atomic resolution. One approach consists of seeding the bilayer by placing individual pre- equilibrated lipid molecules in appropriately chosen locations around the protein structure;[52,77] the membrane is then grown by attaching other lipid molecules to those previously connected to the channel. The second technique consists of generating a protein-shaped cavity in the center of a previously equilibrated lipid bilayer and inserting the transmembrane protein channel into the cavity.

In both approaches a series of equilibration steps based on energy minimization[78] is used to obtain the final configuration of the lipid/protein system. In the latter case, the process does not affect the initial lipid configuration significantly outside the cavity region, and the final configuration of the lipid/

protein system will be very close to the equilibrium structure, thus making it an attractive computational choice.

Several methods to generate a cavity in a lipid bilayer exist. The simplest approach is to remove a cylindrical section of lipid molecules[79] and insert the protein channel into that void. Because lipids are removed in molecular units, the boundary region between the lipid tails and the cylindrical hole tends to be rough. Therefore, many equilibration steps may be required after the protein is inserted, resulting in an unacceptably slow convergence, or in a possibly wrong, unstable, or metastable configuration. An improved approach consists of using a weak cylindrical repulsive force[80] to slowly create the cavity, rather than removing the lipids located inside the region of the cavity. This approach has the benefit of creating a smoother surface at the boundary of the lipid/protein interface. However, the techniques based on approximating the protein molecular surface with a cylinder can result in a difficult equilibration process when non-cylindrical proteins are inserted into the cavity. To address this problem, an arbitrary shaped cavity is obtained by superimposing an atom-size three dimensional grid over the system built by imposing the protein on the lipid patch.[79] All lipid molecules that intersect grid cells containing protein atoms are then removed. Energy minimization steps are subsequently used to further refine the position of the lipids' atoms. Alternatively, a smooth membrane/protein interface with arbitrary geometry can be obtained by applying a weak radial force to create the cavity. In this case, the van der Waals surface[81] of the protein is generated and superimposed on a preequilibrated lipid patch.[82] The lipid atoms inside the van der Waals surface experience an outward radial force, that pushes them out of the cavity. This process is repeated until the hole exactly matches the outer van der Waals surface of the protein.

Time and Space Scale

After choosing an adequate model for each different component of the system and integrating them into a final atomistic model that will be simulated, an important issue is the selection of a discretization scheme to implement the computer representation of the ion channel and its environment. Within the framework of a computer experiment,[83] the adjective *realistic* is strictly related to the phenomena one wants to study, and to the resolution required to reproduce those phenomena. The basic idea for modeling many-body systems is to build a set of rules that apply to each component and let the system evolve dynamically. Ensemble and time averages are then computed to obtain observables that are compared with experiment to validate the model. A characteristic of ion channel systems is that the measurable quantities of direct biological interest evolve in times up to 12 orders of magnitude larger than the smallest atomic or molecular relaxation times (milliseconds versus femtoseconds). In comparison, solid state many-body systems collectively relax in a faster fashion, and the difference between the microscopic

(simulated) and macroscopic (measured) dynamics is four or five order of magnitudes. Extremely slow events, such as charge carrier recombination in semiconductor crystals, also exist in solid state systems, but they can be accounted for in a relatively easy way.

The atomistic representation of a fully hydrated membrane/channel system requires an extremely large number of atoms distributed irregularly on a large computational domain. As an example, consider a K-channel embedded in a POPC membrane as depicted in Fig. 2. The diameter of the selectivity filter in the protein is a few angstroms, and the ionic transit inside the channel occurs in about a microsecond. The selectivity itself is a process that depends on the electrodynamic reaction of the atoms forming the selectivity filter to the electrical and polarization fields due to ions and water inside the filter itself. Given the extremely small distances between the charged components of the system, one expects an extremely rapid relaxation (about ten femtoseconds) that changes the dielectric environment inside the filter. So one needs to simulate the system for at least a few nanoseconds with a resolution of a few femtoseconds in order to observe the transit of one individual ion across the selectivity filter.

Analogously, the channel functionality depends on structural characteristics that extend over a large distance. The interaction of the channel with the membrane is sometimes crucial, both from the structural and electrostatic viewpoint. Furthermore, the structural changes involved in gating are the result of, or are involved directly in, the interaction of the outer protein segments with the membrane. All these facts necessitate the representation of a relatively large system that has to be resolved with angstrom-size accuracy.

Because of the problems related to simulating a large system for a long time with an extremely high resolution, a crucial issue is related to the number of atoms or groups of atoms needed to represent such a system. Indeed, if a brute-force atom-based method is used, the number of individual particles to be simulated is extremely large. In principle, the atomistic representation of a whole protein and of a sufficiently large patch of membrane requires the modeling of at least tens of thousands of particles subjected to a constrained non-local dynamics. Solvation effects must also be accounted for to correctly model both the ion dynamics and the structural properties of the whole system. Biological solutions are typically 0.2 Molar salt but approximatively 55 Molar water. This implies the need for a water model and, consequently, the dramatic increase of the size of the system being simulated.

Experiments

As previously mentioned, a decisive contribution to the understanding of ion channels has been supplied from experiments. The electrical activity of individual channels is measured both in vivo and in vitro under various conditions.[3]

While the description of the experimental setup for channels recording is beyond the scope of this chapter, an important aspect concerning the relevance of the experiments done on channels and their connection with computer modeling should be stressed. Current experiments are not limited to the "simple" observation and characterization of natural channels, but allow for the study of man-made, designed proteins. This capability of building mutant channels by substituting amino acids into the sequence of the natural (or wild) proteins is important for the functional characterization of ion channels, and for the realization of novel macromolecules with specific tunable properties. The need for a strict integration of computational structural chemistry with protein engineering is clear, as well as the need for efficient and reliable computational tools that can direct the experimental work and, at the same time, be validated by it.

ELECTROSTATICS

The channel-membrane-solution system is characterized by an inhomogeneous charge distribution that conditionally allows mobile ions to cross the strong dielectric barrier[84] imposed by the membrane. Therefore, an accurate representation of the electrostatic forces acting on each component is needed to understand the influence of the system's structural properties on its function.

The force fields used by scientists for simulations have been developed with distinct traditions, each appropriate for its own use. In computational chemistry, interest has been in bulk properties of solutions and proteins, in the thermodynamic limit in which boundary conditions do not appear explicitly and where equilibrium (i.e., zero flux of all species) is present. The thermodynamic limit of computational chemistry implies a spatial uniformity of bulk properties that can be analyzed with periodic boundary conditions[85] if the period is longer than the spatial inhomogeneities of the bulk solution.[86]

Contrarily, in computational electronics, the interest has focused on electron devices, which exchange charge with their environment through geometrically and electrically complex boundaries and where internal dielectric discontinuities exist. Simulations are usually performed by varying the applied bias in order to reproduce transient nonequilibrium conditions and to obtain a record of the response of the simulated devices.

Given the substantial differences between the systems being simulated, the force-fields traditionally used by researchers in computational electronics and in chemistry are necessarily different. In particular, short-range coulombic interactions are either neglected in electron device simulations, or they are treated with a stochastic approach[87] rather than deterministically. The same considerations apply for finite size effects.

The highly inhomogeneous charge distribution of ion channel systems makes them closer to electron devices than to bulk homogeneous systems.

This analogy encouraged us to develop the discussion in this tutorial from the viewpoints of both computational electronics and chemistry. The idea here is not to compare the two traditions of computational science, but instead to approach the modeling problem with an interdisciplinary attitude.

Given the complex dynamic properties of the ion channel system, many studies have been performed to examine whether or not any reduced representation could be used to account for some key properties. These attempts have been performed both by theoreticians *and* experimentalists by simulating and measuring properties of simplified systems. From the modeling viewpoint, the evolution of the work of Jordan,[88–91] among others, shows how experimentally obtained structural information has been included into an increasingly complex electrostatic picture as that data became available. Also, the importance of the charge distribution within protein channels is highlighted by the recent work of Varma and Jakobsson,[50] who conducted a systematic study of the ionization states within the lumen of a large porin in order to assess the charge assignment protocols used in the simulation code. Due to the small dimension of ion channels, charges of ions and protein residues are concentrated in small areas. The effects of this "crowded charge" configuration generate extremely localized electric fields that have a significant effect on the polarization state of the system, and perhaps on the molecular structure of the ion channel itself. An example of the effects of closely packed ions in a small region representing a calcium channel can be found in the work of Nonner et al.[92,93] and validated by the equilibrium Monte Carlo simulations of Boda et al.[94–96]

An adequate treatment of the electrostatic properties of the systems of interest is crucial for the understanding of the dynamic properties of ion channels. We now consider the most common methodologies used to implement accurate and efficient electrostatic force-field schemes.

Three efficient approaches for electrostatic modeling of inhomogeneous systems are the fast multipole method (FMM)[97–99] the Ewald summation method,[100] and the Particle–Particle–Particle-Mesh (P^3M) method.[83] Conceptually, these three approaches are very similar[101] because they all consist of writing the total force acting on a charged particle i as the sum of a long-range and a short-range component:

$$\vec{F}_i = \vec{F}_i^{lr} + \vec{F}_i^{sr} \qquad [1]$$

The difference between the three methods is primarily in the calculation of the long-range force $\vec{F}_i^{lr}$. The FMM utilizes a multipole expansion to calculate the long-range force from particles that are far from particle i, while the short-range force is computed through the direct summation of the Coulomb force from particles omitted from the long-range calculation. Within the Ewald method, both the short-range and long-range components are calculated exactly from analytic expressions, where the short-range component is

calculated in real space and the long-range contribution is calculated in reciprocal space. The P^3M formalism accounts for the short-range interactions by directly summing the coulombic particle-particle force in a small volume, while the long-range interaction is determined using the numerical solution of Poisson's equation on a discrete grid over the whole computational domain. Within both the Ewald and P^3M approaches, an overlap between the long-range and short-range domain exists and must be accounted for. This is discussed further in the section "Short-Range Interactions".

It is worth mentioning that in the original work of Hockney and Eastwood[83] on the P^3M approach, the solution of Poisson's equation is calculated in the reciprocal space with Green's functions. In this chapter, an iterative method to calculate the solution of Poisson's equation in real space is discussed. This approach is not commonly adopted for the particle-based simulation of liquid systems. The rather laborious implementation of robust three-dimensional Poisson solvers is probably one of the reasons for the lack of popularity of this approach, which we advocate nevertheless. For this reason, a section of this tutorial is devoted to the discussion of fast iterative methods for the solution of Poisson's equation in position space.

A detailed description of the components of the force $\vec{F}_i$ in Eq. [1] is given in the following sections, for the three different approaches.

Long-Range Interaction

This section is devoted to a discussion of the implementation of the long-range component within the FMM, Ewald summation and P^3M methods.

Multipole Expansion

The FMM[97–99] is based on a multipole series expansion of the long-range potential. The algorithm performance scales linearly with the number of particles,[97] making FMM one of the most efficient approaches available for large systems.

As in the P^3M and Ewald summation methods, within the multipole method formalism the force is separated into a long- and short-range interaction, and the short-range component is resolved through a direct summation over the particle-particle interaction. The long-range component of the force on a generic particle i is computed as

$$\vec{F}_i^{lr} = -q_i \vec{\nabla} \Phi^{lr}(\vec{r}_i) \qquad [2]$$

where the long-range potential $\Phi^{lr}(\vec{r}_i)$ is computed by a pair-wise summation of the charged particles excluded from the direct short-range calculation.

Given a set of point charges $\{j\}$, located inside a sphere of radius R centered about some origin, the long-range Coulomb potential at position $\vec{r}_i$

located outside the sphere (i.e. $|\vec{r}_i| > R$) can be written in spherical coordinates as,

$$\Phi^{lr}(\vec{r}_i) = \frac{1}{4\pi\epsilon_0\epsilon}\sum_{l=0}^{l_{max}}\sum_{m=-l}^{l}\frac{1}{2l+1}\frac{M_{lm}}{r_i^{l+1}}Y_{lm}(\theta_i,\phi_i) \quad [3]$$

where the moments of the expansion are given by

$$M_{lm} = \sum_j q_j r_j^l Y_{lm}^*(\theta_j,\phi_j) \quad [4]$$

and Y_{lm} and Y_{lm}^* are spherical harmonics.[102] A cutoff distance l_{max} has been introduced in Eq. [3] resulting in an error of order $O(r/R)^{l_{max}+1}$.[97] Although the multipole expansion is generally written in spherical coordinates, Cartesian coordinates have also been used for the computation of the potential energy function.[103,104] The representation in spherical coordinates is argued to produce a more efficient implementation than the Cartesian representation.[105,106]

Within the "cellular" version of the multipole method, the computational domain is discretized into a set of rectangular grid cells and the moments of the multipole expansion are computed and stored at each cell. The multipole expansion is only valid for particles that are separated by at least one grid cell,[103] therefore the long-range part of the potential at a position $\vec{r}_i$ is calculated by summing the contributions from all the non-neighboring cells. An improvement of this approach is based on a hierarchy of grids with different cell sizes[107] that allows for the consolidation of cells into progressively larger groups as the distance between the position $\vec{r}_i$ and the cells increase. This coarsening scheme allows for the reduction of the total number of distant cells used in the calculation, and is based on the assumption that the distant charge distribution interacts less intensely than the close one.[105] The accuracy of the calculation remains constant if the ratio between the cell size and the distance is kept constant.[106]

The FMM typically makes use of a local Taylor expansion to further improve the algorithmic efficiency.[97] The difference between the multipole expansion calculated in two different points laying in the same grid cell is assumed to be very small, thus justifying the use of a Taylor series expansion of the potential about the center of the grid cell. The coefficients of the Taylor expansion for each grid cell are calculated once and then evaluated for the position of each individual particle within a given grid cell.

The treatment of boundary conditions can be incorporated in the FMM scheme easily. Periodic boundary conditions as well as Dirichlet, Neumann, and mixed conditions[98] can be accounted for. The FMM approach has been shown to be more efficient than the Ewald summation method (see the next

section), but it results in code that is only faster than P^3M methods for non-physically high numbers of particles.[108] Finally, it is worth noting that the FMM is applicable to all problems involving an r^{-n} pair-wise potential.[98]

Ewald Summation

The Ewald summation method was originally developed as an efficient way to calculate the long-range interactions in ionic crystals,[100] and it has become one of the most common methods for modeling electrostatic properties in periodic structures,[109] particularly for molecular dynamics simulations.[110]

The electrostatic potential energy in a charged system can be written as the summation of all pair-wise coulombic interactions between charges. In a periodic array, it also includes the interaction with the infinite number of replica charges generated by the periodic repetition of the simulated system. The series of coulombic terms converges very slowly, and the solution depends on the order of the summation; i.e., the series is conditionally convergent.

The Ewald formalism is based on a decomposition of the conditionally convergent series into two sums that individually have superior convergence properties. The method involves the addition of an appropriately shaped charge distribution to each charged particle having the same magnitude as the particle but of opposite sign. This charge distribution effectively screens the interactions with neighboring charges, which results in a series that is limited to a short-range domain that in turn makes the resulting summation converge rapidly. To counteract the effects of the artificial charge distribution, a second charge distribution with the same magnitude and same sign as the original point charge is also included for each point charge. If this new charge distribution is smooth, the second summation that accounts for it can be Fourier-transformed and solved efficiently in reciprocal space. From the physical viewpoint, this second summation recovers the long-range interactions that were screened out by introducing the first artificial charge distribution. The two series can then be combined to recover the potential energy caused by the original point charges.

The traditional Ewald summation approach is generally presented in terms of the potential energy of the system. However, the force acting on a given particle is the quantity used by computational approaches, such as molecular dynamics and Brownian dynamics. Therefore, the derivation of the forces (instead of the potential energies) is required, and how these forces are determined is described below.

The exact representation of the long-range component of the force is calculated in the reciprocal space using the Fourier transform. In three-dimensions, the Fourier transform pairs are given by

$$f(\vec{r}) = V^{-1} \sum_{\vec{k}} \tilde{f}(\vec{k}) e^{i\vec{k}\cdot\vec{r}} \qquad [5]$$

and

$$\tilde{f}(\vec{k}) = \int_V f(\vec{r}) e^{-i\vec{k}\cdot\vec{r}} d\vec{r} \quad [6]$$

where $V = L_x \times L_y \times L_z$ is the three-dimensional unit cell in real space, and the components of the vector $\vec{k}$ in the reciprocal space are restricted to the values $k_x = \frac{2\pi l}{L_x}$, $k_y = \frac{2\pi m}{L_y}$, and $k_z = \frac{2\pi n}{L_z}$, where l, m, and n are integers.

The force acting on a charge distribution i can be written as

$$\vec{F}_i^{lr} = -\int q_i S(|\vec{r} - \vec{r}_i|) \vec{\nabla}\Phi(\vec{r}) d\vec{r} \quad [7]$$

where $\vec{r}_i$ is the position of the center of the distribution and S is the shape of the distribution.[102] This integral representation is defined over one real space unit cell, and an extra sum is made over each additional structure to include multiple periodic cells. Within the Ewald approach, the added charge distribution is generally (but not always) modeled with a Gaussian function:[108]

$$S(r) = \left(\frac{\alpha^2}{\pi}\right)^{3/2} e^{-\alpha^2 r^2} \quad [8]$$

where α determines the width of the charge distribution and the Fourier transform of the Gaussian charge distribution is

$$S(|\vec{k} - \vec{k}_j|) = \frac{\alpha^2 V}{\sqrt{8\pi}} e^{i\vec{k}\cdot\vec{r}_j} e^{-k^2/4\alpha^2} \quad [9]$$

The use of a Gaussian distribution is not required, and other functions have been used in the Ewald summation method.[111] For the sake of simplicity, the following derivation is limited to the use of Eq. [8].

The potential can be written in terms of the charge distribution by first applying the Fourier transform to Poisson's equation:

$$k^2 \tilde{\Phi}(\vec{k}) = -\frac{\tilde{\rho}(\vec{k})}{\epsilon\epsilon_0} \quad [10]$$

where the total charge density is given by all remaining charges in the series:

$$\tilde{\rho}(\vec{k}) = \sum_{j \neq i} q_j \tilde{S}(|\vec{k} - \vec{k}_j|) \quad [11]$$

The potential in real space is then written as

$$\Phi(\vec{r}) = -\frac{1}{V\epsilon\epsilon_0}\sum_{\vec{k}\neq 0}\frac{\tilde{\rho}(\vec{k})}{k^2}e^{i\vec{k}\cdot\vec{r}} \quad [12]$$

$$= -\frac{1}{V\epsilon\epsilon_0}\sum_{\vec{k}\neq 0}\sum_{j\neq i}\frac{q_j}{k^2}\tilde{S}(|\vec{k}-\vec{k}_j|)e^{i\vec{k}\cdot\vec{r}} \quad [13]$$

By substituting the equations for the potential into Eq. [7] and integrating over $\vec{r}$, one has

$$\vec{F}_i^{lr} = \frac{q_i}{\epsilon\epsilon_0 V}\sum_{j\neq i}q_j\sum_{\vec{k}\neq 0}\frac{i\vec{k}}{k^2}S(|-\vec{k}+\vec{k}_i|)S(|\vec{k}-\vec{k}_j|) \quad [14]$$

the real part of which gives the final expression for the long-range component of the force:

$$\vec{F}_i^{lr} = \frac{4\pi q_i}{\epsilon\epsilon_0 V}\sum_{j\neq i}q_j\sum_{\vec{k}\neq 0}\frac{\vec{k}}{k^2}e^{-k^2/4\alpha^2}sin(\vec{k}\cdot\vec{r}_{ij}) \quad [15]$$

where $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$.

The conventional Ewald summation method works well for simulations of small periodic systems, but the computation can become prohibitively expensive[112] when large systems are involved, in which the particle number exceeds 10^4. Several numerical techniques have been used to enhance the performance of the traditional Ewald method with mixed results. For example, look-up tables and polynomial approximations[101] have been suggested. The algorithmic performance can also be optimized through the parameter α,[113,114] which determines both the extension of the short-range interaction and the allowable cutoff of the summation over the reciprocal space vectors.[108] Calculating the reciprocal sum is often the most efficient component of the algorithm and α can be chosen to minimize the portion of the summation performed over real space.[113,115] Once the optimal value of α is determined, the performance of the approach can be significantly improved by implementing fast Fourier transform (FFT) algorithms to solve the summation in reciprocal space. The version of the Ewald summation based on these procedures is called the particle-mesh Ewald (PME) method.[112,116] The reciprocal sum is then defined on a discretization grid by using a piece-wise interpolation scheme to assign the charge density to grid points used to evaluate the force (or potential) with FFT.

Poisson Solver in Real Space

Another possible approach for the computation of the long-range force $\vec{F}_i^{lr}$ consists of assigning the charge density to the points of a generally inhomogeneous finite-difference grid, solving Poisson's equation,[102] and differentiating the potential:

$$\vec{F}^{lr}(\vec{r}_p) = -q\vec{\nabla}\Phi(\vec{r}_p) \qquad [16]$$

where $\vec{F}^{lr}(\vec{r}_p)$ and $\Phi(\vec{r}_p)$ represent the force and the electrostatic potential, respectively, at the grid point p located at $\vec{r}_p$. This component of the force also accounts for external boundary conditions, dielectric discontinuities, and fixed charges. The force $\vec{F}_i^{lr}$ on the ion i at the specific position $\vec{r}_i$ is then computed by an appropriate interpolation scheme.

To solve Poisson's equation on a grid, a charge assignment scheme must be devised that builds a charge distribution from the ionic coordinates. Furthermore, once the electrostatic field has been computed on the grid (from the solution of Poisson's equation), the force must be interpolated in each ion location in a way that is consistent with the original charge assignment scheme. In other words, a geometric shape is assigned to each ionic charge though a space-dependent weighting function $W(\vec{r})$,[83] and the geometrical relation between the charge shape and the discretization grid is accounted for in all transformations used to transfer quantities (i.e., charge and force) to and from the mesh centered at $\vec{r}_p$.

The generalized algorithm to accomplish this follows the treatment of Hockney:[83]

1. *Assign charges*:

$$\rho(\vec{r}_p) = \frac{1}{V_p}\sum_i^{N_p} q_i W(\vec{r}_i - \vec{r}_p) \qquad [17]$$

2. *Solve Poisson's equation*:

$$\vec{\nabla}\cdot\epsilon_r\vec{\nabla}\Phi(\vec{r}_p) = -\frac{\rho(\vec{r}_p)}{\epsilon_0} \qquad [18]$$

3. *Calculate electric field*:

$$\vec{E}(\vec{r}_p) = -\vec{\nabla}\Phi(\vec{r}_p) \qquad [19]$$

4. *Interpolate force*:

$$\vec{F}_i^{lr} = \sum_p^{N_p} q_i W(\vec{r}_i - \vec{r}_p)\vec{E}(\vec{r}_p) \qquad [20]$$

where V_p and N_p are the volume of the grid and the number of particles in the grid, respectively. It should be noted that the same function $W(\vec{r})$ must be used both for the charge assignment and for the force interpolation, because the use of a mixed scheme can result in a nonphysical self-force of the particle upon itself. The three most common charge assignment schemes are called the nearest-grid point (NGP), the cloud-in-cell (CIC) and the triangular-shaped cloud (TSC) schemes,[83] and represent the particle as a point charge, a uniformly charged sphere, and a sphere with a linearly decreasing density, respectively. The choice of the weighting function depends on the properties of the system. Once a shape has been chosen for the charge, the corresponding weighting function is determined by the following integral:

$$W(\vec{r} - \vec{r}_p) = \int_{V_p} S(\vec{r}' - \vec{r}) d\vec{r}' \qquad [21]$$

where the function $S(\vec{r})$ represents the shape of the charge ''cloud'' associated with the particle. In one dimension, the weighting functions computed from Eq. [21] are given for the three charge shapes by the following relations:

$$W_{NGP}(x) = \begin{cases} 1 & \left|\frac{x}{H}\right| \leq \frac{1}{2} \\ 0 & \text{else} \end{cases} \qquad [22]$$

$$W_{CIC}(x) = \begin{cases} 1 - \left|\frac{x}{H}\right| & \left|\frac{x}{H}\right| \leq 1 \\ 0 & \text{else} \end{cases} \qquad [23]$$

$$W_{TSC}(x) = \begin{cases} \frac{3}{4} - \left|\frac{x}{H}\right|^2 & \left|\frac{x}{H}\right| \leq \frac{1}{2} \\ \frac{1}{2}\left(\frac{3}{2} - \left|\frac{x}{H}\right|\right)^2 & \frac{1}{2} \leq \left|\frac{x}{H}\right| \leq \frac{3}{2} \\ 0 & \text{else} \end{cases} \qquad [24]$$

where H is the mesh size. For the three-dimensional case, the weighting function is obtained as follows:

$$W(\vec{r}) = W(x)W(y)W(z) \qquad [25]$$

In agreement with the work of Hockney,[83] the TSC weighting function is usually the optimal compromise between accuracy and computational performance for the systems discussed in this chapter.

Using a Poisson solver for the long-range interaction results in two main advantages: (1) the possibility of imposing boundary conditions through externally applied potentials, and (2) the ability to simulate systems with arbitrary ionic concentrations at the boundaries.

Finite-Difference Iterative Schemes

A numerical method is said to be direct when it finds a solution within a given precision and, with a given accuracy, in an initially known number of operations. The time required to solve a differential equation is then well known a priori, and it is independent of the initial or boundary conditions of the problem. Iterative methods, on the other hand, are based on a sequence of approximations to the required solution, starting from an initial guess that converges to the solution. The number of operations, and the time required by these latter methods, are initially unknown because they depend on the initial guess and may vary dramatically as a function of the parameters of the problem.

The self-consistent nature of the simulation approaches described in this chapter requires frequent solutions of Poisson's equation; the potential profiles from one step to the next are very similar to each other because the changes in the charge distribution between two consecutive solutions are very small (but very important for the particle dynamics). The current potential profile can thus generally be used as a good initial guess for the next solution, which makes iterative methods a natural choice within the framework of self-consistent simulation programs. In addition, memory issues[117] (other than pure performance) make the choice of iterative methods appealing in the field of ion channel simulations.

We now present and discuss the basic steps in standard stationary linear iterative methods[118] needed to compute the electrical forces. The current discussion concentrates on the general representation of the two-dimensional Poisson's equation for simplicity

$$\nabla^2 \Phi = f(x, y) \qquad [26]$$

Employing finite differencing on a set of grid points defining the discrete grid denoted by Ω_n, this elliptic differential equation is transformed into an algebraic matrix equation of the form

$$A\mathbf{u} = \mathbf{f} \qquad [27]$$

where the vector $\mathbf{u}$ denotes the solution, the matrix A represents the Laplace operator, and $\mathbf{f}$ is a generic forcing function.

Within the iterative framework, a sequence of approximations $\mathbf{v}^0, \mathbf{v}^1, \ldots, \mathbf{v}^n, \ldots$ to $\mathbf{u}$ is constructed that converges to $\mathbf{u}$.[118] Let $\mathbf{v}^i$ be the approximation to $\mathbf{u}$ after the ith iteration. Because the exact solution $\mathbf{u}$ of Eq. [27] is unknown, one may define the *residual*,

$$\mathbf{r}^i = \mathbf{f} - A\mathbf{v}^i \qquad [28]$$

as a computable measure of the deviation of $\mathbf{v}^i$ from $\mathbf{u}$. Next, the *algebraic error* $\mathbf{e}^i$ of the approximation $\mathbf{v}^i$ is defined by

$$\mathbf{e}^i = \mathbf{u} - \mathbf{v}^i \qquad [29]$$

Subtracting Eq. [28] from Eq. [27] and rearranging terms, it is easily shown that $\mathbf{e}^i$ obeys the so-called *residual equation*,

$$A\mathbf{e}^i = \mathbf{r}^i \qquad [30]$$

Iterative methods can be interpreted as applying a *relaxation operator* to $\mathbf{v}^i$ to obtain a better approximation $\mathbf{v}^{i+1}$ by reducing of the error $\mathbf{e}^i$ related to $\mathbf{v}^i$. In this way, the sequence of approximations $\mathbf{v}^0, \mathbf{v}^1, \ldots, \mathbf{v}^n, \ldots$ is "relaxed" to the solution $\mathbf{u}$.

The expansion of the matrix equation (Eq. [27]) gives the following relation:

$$u_k = \frac{-\sum_{\substack{j=1 \\ j\neq k}}^{n} a_{kj}u_j + b_k}{a_{kk}}, \quad k = 1, 2, \ldots n; \quad a_{kk} \neq 0 \qquad [31]$$

In Jacobi's method,[119] the sequence $\mathbf{v}^0, \mathbf{v}^1, \ldots, \mathbf{v}^n, \ldots$ is then computed by

$$v_k^{(i+1)} = \frac{-\sum_{\substack{j=1 \\ j\neq k}}^{n} a_{kj}v_j^{(i)} + b_k}{a_{kk}}, \quad k = 1, 2, \ldots n; \quad a_{kk} \neq 0 \qquad [32]$$

It should be noted that one does not use the improved values until after a *complete* iteration, within this method. In the closely related Gauß–Seidel method,[118] the values are used as soon as they are computed. One then has

$$v_k^{(i+1)} = \frac{-\sum_{j=1}^{k-1} a_{kj}v_j^{(i+1)} - \sum_{j=k+1}^{n} a_{kj}v_j^{(i)} + b_k}{a_{kk}}, \quad k = 1, 2, \ldots n; \quad a_{kk} \neq 0 \qquad [33]$$

Note that here only one approximation for each v_k needs to be stored at a time. Proofs and discussions about the convergence properties of iterative methods can be found in Young[118] and Dahlquist and Björck.[119]

It is often possible to obtain a substantial improvement of the convergence rate by a simple modification of the Gauß–Seidel method. Note that following the definition of the residual given in Eq. [28], Eq. [33] can be written

as $v_k^{(i+1)} = v_k^{(i)} + r_k^{(i)}$, where $r_k^{(i)}$ is the current residual of the kth equation:

$$r_k^{(i)} = \frac{-\sum_{j=1}^{k-1} a_{kj} v_j^{(i+1)} - \sum_{j=k}^{n} a_{kj} v_j^{(i)} + b_k}{a_{kk}}, \quad k = 1, 2, \ldots n; \quad a_{kk} \neq 0 \qquad [34]$$

The iterative method

$$v_k^{(i+1)} = v_k^{(i)} + \omega r_k^{(i)} \qquad [35]$$

is then the so-called *successive overrelaxation* (SOR) method. Here ω, the *relaxation parameter*, should be chosen so that the rate of convergence is maximized. For $\omega = 1$, the SOR approach reduces to the Gauß–Seidel method. The SOR method has been shown to converge only for $0 < \omega < 2$.[119]

The rate of convergence of SOR is often higher than when using the Gauß–Seidel method, and the additional computational load associated with SOR is negligible. However, the value of ω depends on the grid spacing, the geometrical shape of the domain, and the type of boundary conditions imposed on it.[120] Efforts have been undertaken to find an approach that predetermines the optimal value of ω as a function of the discretization scheme.[121,122] Some improvements in the convergence rate have also been obtained by modifying the processing order of the grid points.[83,123] Despite this modification, the performance of the SOR approach is inadequate for the implementation of real-space Poisson solvers for the simulation of systems discretized on many grid points such as the ones described in this chapter.

The Multi Grid Method

In the previous section, we discussed the basic theory of the classic iterative solution to elliptic problems. The multigrid method allows for a dramatic performance improvement of standard iterative approaches such as the SOR method. The basic principles of its operations are briefly introduced in the following section.

Error Reduction in Classic Iterative Methods. Iterative methods for the solution of large sparse systems of equations have been presented here. These methods produce, by iteration, a sequence of approximations to the required solution, which converge to the solution. This process progressively reduces the error related to each approximation. A given approximation is then accepted as the solution when the deviation from the previous approximation (or some norm of it) is smaller than a predefined threshold. Therefore, an analysis of the error expressed in Eq. [30] as a function of the iteration number (or of the required computer time, because the number of operations per iteration is constant) can provide a useful indication of the solver performance.

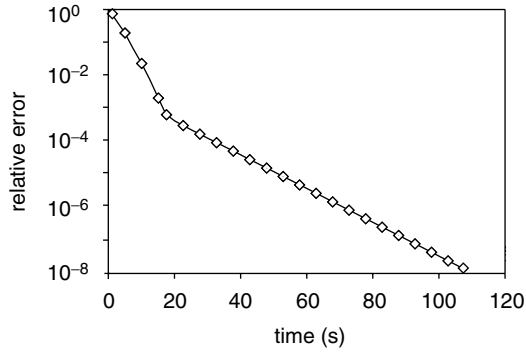


Figure 6 Error reduction rate of the successive overrelaxation method. The smaller slope of the curve for small values of the relative error indicates the poor performance of solvers based on this method.

The absolute value of the maximum relative error is plotted versus the CPU time for an SOR solver in Figure 6. The slope of the curve gives an indication of the performance of the solver: The initial error reduction is very fast, as confirmed by the steep slope of the curve in the upper left corner of the plot. As the error becomes smaller, the slope is less pronounced, which shows a dramatic degradation of the performance. The values of the error that are usually acceptable in particle-based simulations lie in this low-performance region, i.e., typically in the range $[10^{-5}, 10^{-7}]$. The reason for the performance degradation shown in Figure 6 can be understood easily through a spectral analysis of the error before and after a relaxation sweep.

Figure 7 shows, in the upper plots, a schematic representation of the error before (left) and after (right) a single iteration on a unidimensional domain. In the lower plots, the corresponding Fourier components of the error are depicted; in this simplified picture, only two Fourier components are shown. Application of one relaxation sweep affects only the high-frequency

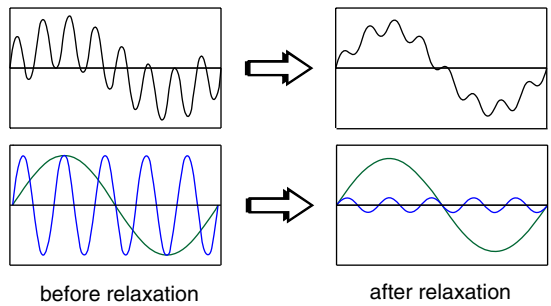


Figure 7 Schematic representation of the relative error of an iterative method after a relaxation sweep. The lower plots show that the low-frequency Fourier component of the error is less reduced than the high-frequency one.

component, which is much more reduced in amplitude than is the low-frequency, long-wavelength component. Thus, the two-slope curve shown in Figure 6 can be explained as follows: The relaxation operator of the iterative method is efficient in reducing only *some* Fourier components of the error. Its error reduction rate slows because the remaining components (the ones with long wavelengths) are not reduced as efficiently. This difference in the error reduction is from the grid spacing: Those components of the error with a wavelength comparable with the grid spacing are reduced more efficiently by the relaxation operator.

Multigrid Basics. The basic idea of the multigrid approach is to simultaneously employ different length scales to efficiently reduce the error. Specifically, one solves Eq. [30] exactly on a grid Ω_{n-1} that is coarser than the initial given grid Ω_n. The resulting value of $\mathbf{e}^i$ is an approximation that is used to correct the previous approximation $\mathbf{v}^i$ that has been determined on the original grid Ω_n:

$$\mathbf{v}^{i+1} = \mathbf{v}^i - \mathbf{e}^i \qquad [36]$$

The advantage of this approach can be understood by considering the Fourier expansion of the error $\mathbf{e}^i$ shown in Figure 7. The long-wavelength components of $\mathbf{e}^i$ are only slightly reduced on the fine grid because their spatial extent exceeds the range of the relaxation operator. The use of a coarser grid renders those components to have an effectively shorter wavelength and thus makes those long-wavelength components "visible" to the relaxation operator. It improves the convergence of the solver dramatically, as compared with a single-grid-based relaxation scheme, such as the SOR.

The simplest version of the multigrid algorithm is the so-called *two-grid iteration* employing only two grid levels. In the ith iteration, the procedure starts from the approximation $\mathbf{v}^i$ of $\mathbf{u}$ in Eq. [27], and the following five steps are performed:

1. Smooth $\mathbf{v}^i$ on the grid Ω_n by applying some suitable relaxation scheme, called *presmoothing*.
2. Compute the residual according to Eq. [28] and transfer it to the coarser grid Ω_{n-1}. This step is called *restriction*.
3. Solve Eq. [30] exactly on the grid Ω_{n-1}.
4. Interpolate the resulting $\mathbf{e}^i$ to the finer grid Ω_n. This step is called *prolongation*. Subsequently, calculate $\mathbf{v}^{i+1}$ from Eq. [36].
5. Smooth $\mathbf{v}^{i+1}$ on the grid Ω_n by applying some relaxation method, called *postsmoothing*.

It is possible to extend the two-grid algorithm to a sequence of grids that are increasingly coarse, because Eq. [30], applied on the grid Ω_{n-1}, has *the same form* as Eq. [27] on Ω_n. It is achieved by recursively applying the complete algorithm (steps 1 through 5) at step 3. The recursion scheme is

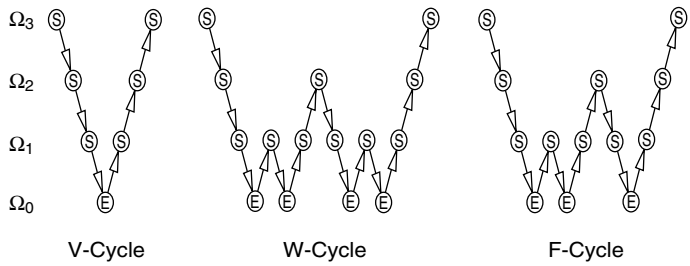


Figure 8 Standard multigrid structure for a V-, W-, and F-cycle. In this figure, the number of grid levels is four, the circles represent a Smoothing operation or an *E*xact solution, and the arrows indicate prolongation (upward) and restriction (downward) operations. Ω_0 represents the coarsest and Ω_3 the finest grid.

stopped when the coarsest grid Ω_0 is reached. At that grid level, Eq. [30] is solved exactly. Because this grid usually contains only a few points, it can be done easily. This multiscale algorithm defines one complete *multigrid iteration* (labeled by the superscript i). The whole procedure is then repeated until the required convergence threshold is reached.

This discussion of the multigrid iteration refers to a cyclic structure called the V-cycle (see Figure 8). More generally, one may define the multigrid iteration as the recursive application of γ two-grid cycles at any grid level, the V-cycle being characterized by $\gamma = 1$. The case $\gamma = 2$ is called W-cycle. It is possible to use any number γ of two-grid cycles at each level, obtaining better convergence at the cost of increased complexity of the algorithm. A special cycle, the so-called F-cycle, is also shown in Figure 8. It is important because its structure often optimizes the tradeoff between pure performance and complexity. The F-cycle on Ω_i is recursively defined as follows:[124] Its coarse grid part consists of an F-cycle on Ω_{i-1}, followed by a V-cycle on Ω_{i-1}. An F-cycle on the coarsest grid Ω_0 is just a V-cycle.

Finally, it should be noted that the multigrid method can be used as either an iterative process or as a direct solver (the so-called *full multigrid* or *nested iteration* method[125]).

Algorithmic details on the multigrid method can be found in the excellent works of Hackbusch[125] and Brandt.[126,127] It should be noted that the multigrid approach can be easily applied to adaptive non-tensor-product grids,[126,127] allowing for variable resolution in regions of the computational domain where the charge concentration is high. A discretization scheme based on adaptive grids can result in a further increase in performance when simulating highly inhomogeneous systems such as biological membranes or complex proteins.

The SOR method for solving Poisson's equation in ion-channel applications is not advocated here because of its slower convergence compared with the multigrid approach and because of its inefficiency for large problems (Figure 9). It is recognized, however, that the extreme simplicity of the SOR

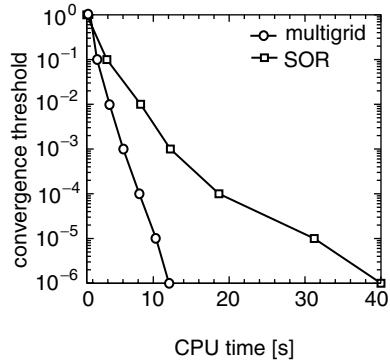


Figure 9 Comparison of the CPU time required to solve Poisson's equation with the multigrid and SOR method. The computational domain consists of a $65 \times 65 \times 65$ homogeneous mesh.

algorithm makes it an attractive choice. A typical SOR solver can be implemented with a few tens of lines of code, whereas our three-dimensional multigrid solver is several thousand lines long.

Short-Range Interaction

The short-range force is written as the sum of three terms:

$$\vec{F}_i^{sr} = \vec{F}_i^{C} + \vec{F}_i^{W} + \vec{R}_i \qquad [37]$$

where $\vec{F}_i^{C}$ is the coulombic force from all particles within a predefined short-range domain, $\vec{F}_i^{W}$ represents the effects of the van der Waals forces, and $\vec{R}_i$ is a "reference force"[83] that corrects the double counting of charges caused by the overlap between the short-range and long-range domains occurring in both the Ewald and the P^3M methods. No overlap exists within the FMM formalism, so the reference force is null in such a scheme. The forces in Eq. [37] are expressed as follows:

$$\vec{F}_i^{C} = \sum_{j \neq i}^{\Lambda_i} \frac{q_i q_j}{4\pi\epsilon_r\epsilon_0 |\vec{r}_i - \vec{r}_j|^2} \hat{r}_{ij} \qquad [38]$$

$$\vec{F}_i^{W} = \begin{cases} \sum\limits_{j \neq i}^{\Lambda_i} \frac{24\epsilon_{ij}}{|\vec{r}_i - \vec{r}_j|} \left[2\left(\frac{\sigma_{ij}}{|\vec{r}_i - \vec{r}_j|}\right)^{12} - \left(\frac{\sigma_{ij}}{|\vec{r}_i - \vec{r}_j|}\right)^{6} \right] \hat{r}_{ij} & \textit{Lennard-Jones} \\ \sum\limits_{j \neq i}^{\Lambda_i} \frac{\beta_{ij} |q_i q_j|}{4\pi\epsilon |\vec{r}_i - \vec{r}_j| (p+1)} \left(\frac{s_i + s_j}{|\vec{r}_i - \vec{r}_j|}\right)^{p} \hat{r}_{ij} & \textit{inverse power} \end{cases} \qquad [39]$$

$$\vec{R}_i = -\sum_{j \neq i}^{\Lambda_i} \frac{q_i q_j}{4\pi\epsilon_r\epsilon_0} \iint S(\vec{r}_1) S(\vec{r}_2 - \vec{r}_{ij}) \frac{(\vec{r}_1 - \vec{r}_2)}{|\vec{r}_1 - \vec{r}_2|^3} d\vec{r}_1 d\vec{r}_2 \qquad [40]$$

where Λ_i is the domain of the short-range interaction (see below), ϵ_r is the relative dielectric constant, ϵ_0 is the permittivity of vacuum, q is the charge, and $\vec{r}_{ij}$ is the distance between ions.

The van der Waals force $\vec{F}_i^W$ is often modeled with the Lennard–Jones function or by an inverse power relation.[128] The former is based on the two fitting parameters σ_{ij} and ϵ_{ij}, representing, respectively, the maximum attraction distance and the strength of the interaction.[129] For ions of different species, the Lennard–Jones parameters are typically calculated by combining the values of the individual species:[129]

$$\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j) \quad \text{and} \quad \epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j} \qquad [41]$$

In the expression of the inverse power law, β_{ij} is an adjustable parameter, s_i is the radius of the ith particle, and p is a hardness parameter that also represents the interaction strength. A comparison of the interionic potential profile for the two different pair potential schemes in an aqueous KCl solution is shown in Figure 10. The parameters used for the short-range potentials are taken from Im et al.[130] for the Lennard–Jones function and from Hockney[83] for the inverse power relation.

The final component of the particle–particle force is the reference force $\vec{R}_i$, which depends on the shape S of the ionic charge. As stated, within the P^3M approach, the particle–particle portion of the force is calculated for ions within the relatively small spherical region Λ_i. The role of the reference force is to correct for the overlap between Λ_i and the entire system over which the mesh force $\vec{F}_i^{lr}$ is calculated. In other words, the sources of the electrostatic force acting on a given charged particle are classified as "far sources" (including boundary conditions) that are accounted for efficiently by the solver for the long-range interaction, and "close sources" generating forces that are not

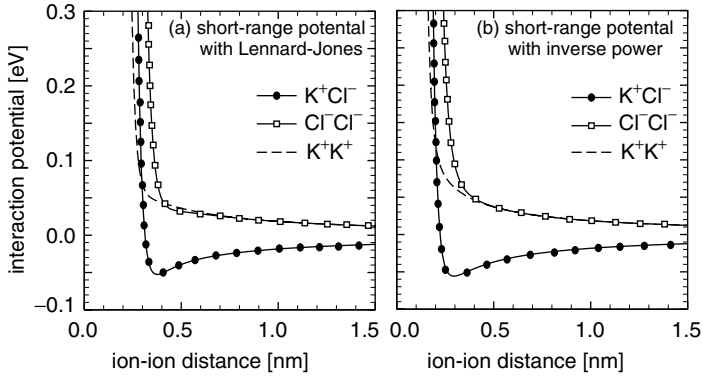


Figure 10 Comparison of short-range Lennard–Jones and inverse-power potential for K^+ and Cl^- in aqueous solution.

resolved by the solver and must be computed by the more CPU-expensive particle–particle scheme. The domain Λ_i defines the high-resolution region around a given ion. For obvious reasons, the computation of the long-range interactions cannot be obtained by subtracting the charges within Λ_i, it would indeed require a full solution for each particle at each iteration; so the effect of those sources is subtracted from the potential distribution after the solution has been obtained. This correction is accomplished by the reference force.

Clearly, the size of the region Λ_i should be chosen as small as possible based on performance considerations. The key aspect that limits the minimum size of Λ_i is the size of the ionic charge used for the charge assignment scheme (see "Poisson Solver in Real Space"). As stated, the charge distribution is computed by assigning a "cloud" of charge to each ion. The cloud has a specific geometric shape and a predefined charge density. When calculating the total force on a given ion i, all charged particles $j \neq i$ whose charge cloud is overlapping with that of i are considered "close sources" of the electrostatic force, and must be included in the domain Λ_i.

For example, if the chosen ionic electrostatic shape S is a sphere with a uniformly decreasing charge density, the corresponding weighting scheme is the TSC:[83]

$$S(r) = \begin{cases} \frac{3}{\pi r_c^4}(r_c - r) & r_c \leq r \\ 0 & \text{else} \end{cases} \qquad [42]$$

where r_c is the radius of the spherical charge cloud. In this case, the natural choice for the minimum cutoff radius that defines the short-range region Λ_i is twice r_c. The reference force is then found analytically by substituting the shape function $S(r)$ into Eq. [40]:

$$R(r) = \frac{q_i q_j}{4\pi\epsilon_r\epsilon_0} \begin{cases} \frac{4}{35r_c^2}(224\zeta - 224\zeta^3 + 70\zeta^4 + 48\zeta^5 - 21\zeta^6) & 0 \leq \zeta \leq 1 \\ \frac{4}{35r_c^2}(12/\zeta^2 - 224 + 896\zeta - 840\zeta^2 + 224\zeta^3 & \\ \qquad +70\zeta^4 - 48\zeta^5 - 7\zeta^6) & 1 \leq \zeta \leq 2 \\ \frac{1}{r^2} & \text{else} \end{cases} \qquad [43]$$

where $\zeta = r/r_c$. To reduce the computational burden, the reference force is tabulated as a function of the distance between ion pairs as suggested by Hockney[83] and subsequently by Wordelman[131] during initialization.

The components of the force between an anion and a cation inside the short-range domain ($2r_c = 2$ nm) are shown in Figure 11 as a function of the interionic separation. The two ions are placed in a 500-mM KCl solution, with no external bias. As expected, the reference force and mesh force have the same amplitude and therefore will cancel within the short-range domain.

Within the Ewald approach, the charge distribution is defined as a Gaussian function (see Eq. [8]), and the sum of the direct Coulomb force

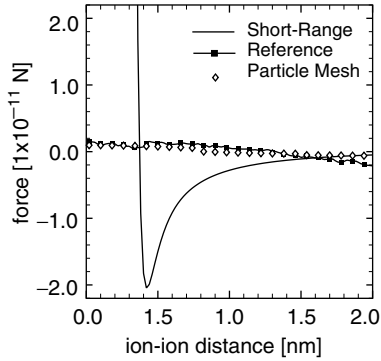


Figure 11 Components of the force inside the short-range domain calculated between two ions of opposite charge in a 500-mM solution of KCl with no bias.

and reference force can be written as an analytic expression. The final expression for the short-range interaction is then

$$\vec{F}_i^{sr} = \vec{F}_i^{W} + \frac{q_i}{4\pi\epsilon\epsilon_0}\sum_{j\neq i} q_j\left(\mathrm{erfc}[\alpha|\vec{r}_{ij}|] + \frac{2\alpha}{\sqrt{\pi}}|\vec{r}_{ij}|e^{-\alpha^2 r_{ij}^2}\right)\frac{\vec{r}_{ij}}{|\vec{r}_i - \vec{r}_j|^3} \qquad [44]$$

Boundary Conditions

Once the significant components of the system have been chosen, a computational domain is then defined to enclose them. The geometry of the simulation box must define a volume that realistically encloses the physics of the system, with boundary conditions mimicking the effects of the larger, real system being modeled. Within the ion channel framework, only a small fraction of the cellular lipid membrane is simulated; thus, the dimension of the computational domain is minimized to reduce the computational burden. Consequently, the boundary conditions must be chosen carefully so that unwanted computational artifacts are not introduced into the simulation results.

The most popular, and somehow elegant, choice is to impose periodic boundary conditions on a parallelepiped-shaped domain. This approach is adequate for simulating bulk systems because it ensures continuity of the ionic flux and of the force field at the boundaries. It is also compatible with the algorithm that accounts for long-range electrostatic interactions in the Ewald summation method. However, the periodic boundary approach also has drawbacks that are sometimes difficult to address. The main problem involves the charge distribution within the computational domain. The source of this problem is from the highly inhomogeneous ion charge distribution that generates far-reaching electric fields. When a periodic boundary "cuts" the field distribution, significant perturbations are generated in the forces that drive the

dynamics of the system; care must be taken in choosing the size of the periodic box. To validate the results obtained with periodic boundaries, Yang et al. suggest running the same simulation on computational domains of different size; that way the presence of size-dependent artifacts can be deleted and then excluded.[132] Furthermore, periodic boundaries make it extremely difficult to simulate systems with inhomogeneous charge distributions at the boundary, such as those systems with different dielectric coefficients in different regions or with different solutions on either side of the membrane. Also, the common experimental practice of applying external potentials across the solution is difficult to reproduce in a periodic system.

The use of nonperiodic boundary conditions is also complicated, especially by the necessity for having a mechanism that effectively and realistically recirculates mobile components (ions and sometimes water molecules) that escape from the computational domain. The injection scheme is trivial in periodic systems, but it is not at all obvious[133] for nonperiodic systems.

Two main types of electrostatic boundary conditions are used in nonperiodic systems. The Dirichlet boundary condition fixes the value of the electrostatic potential, whereas the Neumann method sets the value of the normal component of the electric field.[83] One approach employed to regulate the injection of ions in nonperiodic systems is to use reservoirs of particles and a simple stochastic boundary that maintains a given concentration value in the entire system.[134] Ions are recycled from one side of the domain to the other whenever there is an imbalance caused by a conduction event. It has been shown that the simple stochastic boundary method of constant injection gives very similar fluctuations in the particle number of regions of the computational domain far from the injecting boundaries.[134] This approach is simple and efficient, but it has the drawback of not being able to handle concentration gradients. Rather than maintaining the concentration in the entire domain, another approach consists of simply fixing the concentration in the Dirichlet boundary cells, and injecting particles to sustain this concentration. That way, the concentrations at different boundaries can be different.

A third approach[130] is to inject particles based on a grand canonical ensemble distribution. At each predetermined molecular dynamics time step, the probability to create or destroy a particle is calculated and a random number is used to determine whether the update is accepted (the probability for both the creation and the destruction of a particle must be equal to ensure reversibility). The probability function depends on the excess chemical potential and must be calculated in a way that is consistent with the microscopic model used to describe the system. In the work of Im et al.,[130] a primitive water model is used, and the chemical potential is determined through an analytic solution to the Ornstein–Zernike equation using the hypernetted chain as a closure relation.[72] This method is very accurate from the physical viewpoint, but it has a poorer CPU performance compared with simpler schemes based on

constant injection rates because of the continuous calculation of the chemical potential.

PARTICLE-BASED SIMULATION

A key component of particle-based simulation methods involves the coupling of the dynamics of the charge carriers (ions) with the field of forces generated by the external boundary conditions as well as by the internal electrostatic interactions between the components of the system. This self-consistent coupling approach has been successfully employed for more than three decades in plasma simulations.[135] The adjective self-consistent refers to the fact that the forces caused by the electrostatic interactions within the components of the system depend strictly on the spatial configuration of the components and must be updated continuously as the dynamics of the system evolves.

Self-consistency is achieved by periodically "freezing" the dynamics and by updating the spatial force distribution. The dynamics is then resumed in the "updated" field of forces, which is assumed to be constant for a time Δt that, in the cases of interest, is usually on the order of a femtosecond. At the end of Δt, a new field is computed from the new charge distribution.

The need for self-consistency between charge and force distributions is caused by the spatial inhomogeneities of the systems under scrutiny. The long-range nature of the electrostatic interaction makes the relation between ionic concentration and field distribution highly nonlinear.[136,137] Significant differences in methodology for implementing the potential functions in simulation programs[138] exist. Figure 12 depicts the flowchart of a typical particle-based algorithm. The self-consistent aspect of the approach is enforced within the main iteration cycle, where the field of force and the ionic dynamics are continuously coupled during the simulation.

The following section is devoted to two popular approaches for particle-based simulations of ionic charge transport in transmembrane proteins. The name used for the family of approaches to be discussed has its origin in the fact that at least some components of the system are represented as computer "particles" and their trajectories are tracked in phase-space. Although mobile ions in solution are always modeled as particles, their dynamics can be Brownian or Newtonian based on the representation of the water solvent. When the effects of water on the system dynamics are modeled through macroscopic quantities such as the diffusion coefficient or the dielectric constant rather than by treating each atom (or collection of atoms) as a unique particle that exerts its influence on the system, we say that the solvent is treated implicitly; i.e., we are implying in some way that the water is there influencing the system's dynamics. Alternatively, the solvent model is defined as being "explicit" if the water molecules are represented as separate entities, each obeying the laws of physics and thus influencing the system's dynamics.

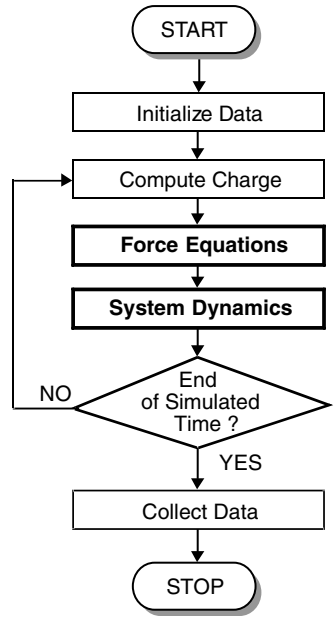


Figure 12 Flowchart of the self-consistent, particle-based algorithm.

Implicit Solvent: Brownian Dynamics

When the solvent is treated as a continuous dielectric background that interacts stochastically with the mobile ions, the ionic trajectories can be modeled with the Langevin formalism.[139,140] In particular, the strict or full Langevin equation can be used, which assumes Markovian random forces and neglects correlations (both spatially and temporally) of the ionic motion:

$$m_i \frac{d\vec{v}_i(t)}{dt} = -m_i \gamma \vec{v}_i(t) + \vec{F}_i(\vec{r}_i(t)) + \vec{B}_i(t) \qquad [45]$$

where m_i is the reduced mass of the ith ion, $\vec{v}_i(t)$ is its velocity at time t, γ is the friction coefficient (i.e., the inverse of the ionic velocity relaxation time), $\vec{F}_i$ is the force on ion i caused by all other particles in the system and boundary conditions (including internal dielectric discontinuities), and $\vec{B}_i$ is a fluctuating force that mimics the molecular bombardment of water on the ion and is modeled with a Markovian random variable. The fluctuating force can, therefore, be written explicitly as

$$\vec{B}(t) = \sqrt{2\gamma m k_B T}\dot{w} \qquad [46]$$

where a Gaussian white noise term is given by $\dot{w}$.

The Langevin equation is discretized temporally by a set of equally spaced time intervals. At predetermined times, the ion dynamics is frozen, and the spatial distribution of the force is calculated from the vector sum of all its components, including both the long-range and the short-range contributions. The components of the force are then kept constant, while the dynamics resumes under the effect of the updated field distribution. Self-consistency between the force field and the ionic motion in the phase space is obtained by iterating this procedure for a desired amount of simulation time. The choice of the spatial and temporal discretization schemes plays a crucial role in computational performance and model accuracy.

The integration scheme used for Eq. [45] is chosen based on fulfilling two requirements: maintaining energy stability and allowing for large time steps. The latter requirement is related to the need to investigate system properties for the typically long biological time scales, which can be on the order of microseconds or more. Using long time steps reduces the number of operations for each unit of simulated time, thus increasing the performance of the simulation code. Counteracting this is the requirement that the time step must be small compared with the mean time between particle collisions. An excessively coarse time discretization would not account for rapid variations in the short-range force, and it does not correctly account for its coulombic singularity. A large time step typically results in a spurious heating of the particle ensemble that then becomes energetically unstable.[83]

Two common implemented integration schemes for the Langevin equation are the standard first-order Euler scheme and the Verlet-like method of van Gunsteren and Berendsen.[141] The latter is a third-order model that reduces to the Verlet algorithm[142] when the friction coefficient in the Langevin equation is zero (see section on Explicit Solvent below). This approach by van Gunsteren and Berendsen allows for a larger time step as compared with the Euler method. Both schemes are discussed in the following sections, and a comparison is offered.

Particle Tracking: Euler Integration

The first-order Euler integration scheme reduces the Langevin equation to

$$\vec{v}_i(t+\Delta t) = \vec{v}_i(t) - \Delta t\left[\gamma\vec{v}_i(t) - \frac{\vec{F}_i}{m_i} - \sqrt{\frac{2\gamma k_B T}{m_i \Delta t}}\vec{N}(0,1)\right] \qquad [47]$$

where Δt is the integration time step and $\vec{N}(0,1)$ is a three-dimensional Gaussian random variable with zero mean and a variance of 1. The spatial trajectories are calculated with Newtonian mechanics. To represent the fluctuating force as a stationary Markovian, Gaussian process, the time-step duration Δt must be much smaller than the reciprocal of the friction coefficient γ in the Langevin equation (Eq. [45]).[141] It results in a fine (and

computationally expensive) time discretization when ionic solutions are simulated.

Particle Tracking: Verlet-like Integration

The need for carrying out impractically short time steps was addressed by van Gunsteren and Berendsen[141] who accounted for the evolution of the fluctuating force during the integration time step. In their method, the force on the ith particle at time t_{n+1} is first expanded in a power series about the previous time t_n

$$F_i(t_{n+1}) \sim F_i(t_n) + \dot{F}_i(t_n)(t_{n+1} - t_n) \tag{48}$$

where $\dot{F}$ denotes the time derivative. The power series expansion is then substituted into Eq. [45] and the resulting solution of the Langevin equation is

$$\begin{aligned} v_i(t_{n+1}) = {} & v_i(t_n)e^{-\gamma\Delta t} + (m_i\gamma)^{-1}F_i(t_n)(1 - e^{-\gamma\Delta t}) \\ & + (m_i\gamma^2)^{-1}\dot{F}_i(t_n)(\gamma\Delta t - (1 - e^{-\gamma\Delta t})) \\ & + (m_i)^{-1}e^{-\gamma\Delta t}\int_{t_n}^{t} e^{-\gamma(t'-t_n)}B_i(t')dt' \end{aligned} \tag{49}$$

where $\Delta t = t_{n+1} - t_n$ is the integration time step. Note that the fluctuating force $B_i(t)$ is retained inside the integral. The ion's position is calculated with the expression

$$\begin{aligned} x_i(t_{n+1}) = {} & 2x_i(t_n) - x_i(t_{n-1})e^{-\gamma\Delta t} \\ & + \int_{t_n}^{t_n+\Delta t} v_i(t')dt' + e^{-\gamma\Delta t}\int_{t_n-\Delta t}^{t_n} v_i(t')dt' \end{aligned} \tag{50}$$

and, finally, the updated particle position is written as

$$\begin{aligned} x_i(t_{n+1}) = {} & x_i(t_n)[1 + e^{-\gamma\Delta t}] - x_i(t_{n-1})e^{-\gamma\Delta t} \\ & + (m_i\gamma)^{-1}F_i(t_n)(\Delta t)[1 - e^{-\gamma\Delta t}] \\ & + (m_i\gamma^2)^{-1}\dot{F}_i(t_n)(\Delta t)[0.5\gamma\Delta t(1 + e^{-\gamma\Delta t})] \\ & - [1 - e^{-\gamma\Delta t}] + X_i^n(0, \Delta t) + e^{-\gamma\Delta t}]X_i^n(0, -\Delta t) \end{aligned} \tag{51}$$

where

$$X_i^n(0, \Delta t) = (m_i\gamma)^{-1}\int_{t_n}^{t_n+\Delta t}[1 - e^{-\gamma(t_n+\Delta t-t')}]B_i(t')dt' \tag{52}$$

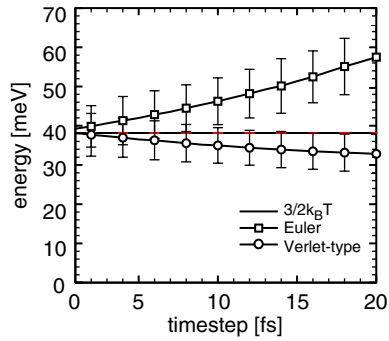


Figure 13 Steady-state energy of an ensemble of anions and cations in a 150-mM solution of KCl as a function of time step, for both the Euler and the Verlet-like integration schemes.

Equation [52] is also a Markovian stochastic process with zero mean and variance Δt. The quantity $X_i^n(0, -\Delta t)$ is correlated with $X_i^{n-1}(0, \Delta t)$ through a bivariate Gaussian distribution. In the zero limit of the friction coefficient, this set of equations corresponds to the trajectories obtained with the Verlet algorithm.[141]

The set of trajectories resulting from the Verlet-like integration scheme as compared with the Euler scheme is not limited by the velocity relaxation time, and consequently a longer time step can be used. Figure 13 shows a plot of the steady-state average ionic energy versus time-step interval for a 150-mM KCl solution simulated for 1 ns in the absence of an external electric field. The Euler and Verlet-like algorithms give similar results for time steps below approximately 10 fs, but larger time steps result in a greater energy drift for the Euler integration scheme.

Explicit Solvent: Molecular Dynamics

The molecular dynamics approach allows for the simulation of the system components *individually* with atomic resolution. Broadly speaking, an appropriately constrained Newtonian dynamics is used to capture the evolution of particles representing individual ions, atoms, or groups of atoms in the force field generated by electrostatic and van der Waals interactions together with boundary conditions. One difference between molecular dynamics and Brownian dynamics is the way the solvent is modeled: Water molecules are typically treated explicitly within the molecular dynamics framework.

The role of water in ion permeation through narrow channels was stressed previously; a model that accounts for the dynamics of the ionic solvation state is needed for a full understanding of channel functionality. Furthermore, the atomic resolution of molecular dynamics includes sufficient information to (in principle) treat polarization effects with highly accurate,

microscopic resolution. It is definitely an advantage of molecular dynamics over Brownian dynamics, which tracks the individual ionic trajectories on the atomic scale, but uses blurred-out collective properties such as the dielectric constant or the friction coefficient to express the interaction of the ions with their environment. Unfortunately, the computational burden associated with molecular dynamics simulations of ion channels is such that only relatively small systems can be simulated for times that are too short to produce statistically relevant estimates of macroscopic observables such as the ionic current flowing through an open channel.[11,143] Although it is obvious that the macroscopic parameters "friction coefficient" and dielectric coefficient will not capture the atomic detail important for ion movement or permeation in Brownian dynamics, it is not obvious that simulations of ion channels will reproduce the bulk properties of friction or dielectric response with explicit molecular dynamics methods until the simulations are actually performed and the results compared with the experiment. Calibration of equilibrium systems with atomic detail form the basis of equilibrium molecular dynamics and Monte Carlo simulations of ionic solutions.[95,96]

Many models are used to include the microscopic effects of water molecules on biological systems, and most of them are based on parameterized force field schemes that are tuned to reproduce some bulk macroscopic properties of the solvent. For a given system, the choice of a specific water model is based on the usual tradeoff between accuracy and computational complexity. Furthermore, even if a particular model fits a type of data better than another—for example, dielectric constant better than density versus temperature—the choice of which model to use is not obvious.

Water models used in ion channel simulations must reproduce, among other things, the solvent structural properties measured by the radial distribution function (RDF), mass transport characteristics like the diffusion coefficient, and the macroscopic polarization behavior, such as the dielectric constant. These models should also account for the local interactions of the water with molecules in the protein structure. It is especially important because polarization effects may play a role in the ion permeation process. It should also be noted that the bulk ionic concentrations of biologically relevant systems are relatively low, so usually a large number of solvent molecules must be simulated to ensure the presence of a statistically relevant number of ions within the simulation domain. Indeed, ions in concentrations of 10^{-6} M often control biological reactions of great importance: Biochemistry textbooks pay much attention to the cofactors or coenzymes that control life's metabolism. The effects of these cofactors depend heavily on concentration. Thus, simulations must be able to estimate accurately the activity (i.e., effective concentration) of such trace ions if they are important in the system being studied. Consequently, the greatest number of atoms in molecular dynamics simulations is usually those of the water molecules, adding the computational efficiency as a final stringent requirement for implementing a particular solvent

model. It is difficult to calculate more than a few nanoseconds of simulation time in ion channel simulations, and in fact only Crozier et al.,[144,145] to the best of the author's knowledge, has been able to compute ion trajectories for a microsecond.

Rigid, fixed-charge water models are widely used in molecular dynamics simulations.[146] Their popularity is from their algorithmic simplicity and from their ability to reproduce many thermodynamic properties that match experiment. Within these models, point charges combined with empirical potentials are used to model the electrostatic interaction of the water molecule[147] with its environment. The charges are placed at specific sites within the molecular volume,[147] and the effective potentials are tuned to reproduce the average (bulk) effects of polarization.

Among these approaches, 3-, 4-, and 5-site models have been implemented, with different geometric configurations. Within the simple 3-site model, the positions of three charges are set to the sites of the hydrogen and oxygen atoms, whereas the negative charge is moved from the oxygen site toward the hydrogens along the bisector of the hydrogen–oxygen–hydrogen angle[146] in the 4-site representation. For the 5-site model, discrete charges are located at the positions of the hydrogen atoms, and an additional lone pair is oriented tetrahedrally around the oxygen. The number of operations in the molecular dynamics algorithms scales with the square of the number of interaction sites, thus explaining the popularity of the 3- and 4-site water models.

The family of 3- and 4-site models include the simple point charge model (SPC) and the transferable intermolecular potential functions (TIPS). The SPC is a basic 3-site model[148] with parameters adjusted to reproduce the energy and pressure of liquid water under ambient conditions. Parameters are further optimized to fit structural properties, specifically the second peak of the RDF of the oxygen atoms. The TIPS started as a 3-site liquid-phase model that was later extended to a 4-site (TIP4P) configuration[149] to reproduce the second peak structure[149] of the oxygen RDF. The TIPS- and SPC-based models are considered to be the most efficient because they require the lowest number of interaction calculations while providing accurate estimates of the intermolecular energy and density.[144] An extended version of the SPC fixed-charge water model was developed, which included the polarization through a mean-field description of the induced moments (SPC/E[150]). This model provides a more accurate RDF and an improvement in the calculated diffusion coefficient as compared with the standard SPC model. The parameters used in this approach are still empirical, and they are adjusted to fit known physical properties. The inclusion of a realistic self-consistent description of polarization in water models is currently an important research topic.

The integration schemes used for Newtonian dynamics are simpler than that employed in the Brownian dynamics simulation based on Langevin's equation (see the section "Implicit Solvent: Brownian Dynamics"). A popular choice[11] for Newtonian molecular dynamics is the Verlet integration scheme

that conserves volume in phase space and is therefore symplectic.[151] To reduce the number of computations per unit of simulated time, much work has been devoted to integration schemes with variable time resolution.[152,153] The idea behind these multiple time step methods is that a lower time resolution is required for assessing the long-range components of the force field, which translates into computing the long-range forces less frequently than the short-range forces.[154]

Calculation of Free Energy

Despite the limitations discussed, molecular dynamics simulations supply invaluable insight into channel functionality because (1) they allow for a microscopic analysis of the structural fluctuations of the channel-membrane system,[52,155] and (2) they allow for a mapping of the energetics of the ion permeation process in terms of the potential of mean force,[156] which represents the free energy content of the system as a function of a reaction coordinate.[157] The free energy landscape associated with ions in the proximity of a channel can then be used as input for faster and less detailed simulation tools, such as the Brownian dynamics approach.[158] Analogously, the ion diffusion coefficient within the channel can be extracted from molecular dynamics calculations[159,160] and used with electrodiffusive continuum models (see the next section on Flux-Based Simulation).

Indeed, all ingredients for a complete thermodynamic characterization of the system are available in molecular dynamics simulations: atomic resolution, protein flexibility, membrane fluctuations, explicit solvent, and ionic motion. Because the free energy profile controls ion conduction,[161] along with nonequilibrium parameters like the diffusion coefficient, one can expect to fully understand the permeation (and selectivity) processes from it. Furthermore, because one can explore the energetics of molecular configurations in response to external stimuli, free energy calculations can in principle supply information about gating mechanisms or, at least, could be used to confirm hypotheses derived from indirect experimental observations.

The task of computing free energy from molecular dynamics trajectories can be difficult because of the (non)statistical relevance of the trajectories and the inaccuracy of the force field (discussed in the next section). Extracting statistically homogeneous data from raw molecular dynamics simulations of ion channels is arduous. The highly inhomogeneous charge distribution generates a bumpy electrostatic landscape for the ions' dynamics. Consequently, regions of the conduction path that are needed for permeation are rarely visited by ionic trajectories.[143] Fortunately, much work has been done to enhance the statistical relevance of low-occupancy regions, and several numerical techniques have emerged as being highly effective in sampling those regions. For example, the addition of an artificial restraint on the ionic trajectories artificially biases them toward more sparsely populated regions, increasing the accuracy of the free energy profile. The effects of this external perturbing

potential are then processed out of the calculated results, giving a statistically enhanced free energy profile. This technique, called umbrella sampling,[162] can be used simultaneously on different regions of the same reaction coordinate, or on a multidimensional reaction space (see Chapter 9 in Becker et al.[163]). This process allows one to obtain an accurate free energy profile by means of loosely coupled simulations that can be run concurrently on separate computers. Umbrella sampling has been used to study the multi-ion free energy profile of the selectivity channel of KcsA,[156] where the existence of binding sites on the extracellular side of the channel were predicted. These predictions were subsequently confirmed by high-definition experimental observations.[41]

The Force Field

If the ionic trajectories can be statistically enhanced by using appropriate computational techniques like umbrella sampling, it is imperative that we also increase the accuracy of the forces being computed in particle-based simulations. State-of-the-art simulation packages use the force field decomposition discussed in the section on electrostatics (see Eq. [1]), where either the long-range component of the force is neglected or is included in the Ewald approach. Therefore, most simulations including long-range electrostatic interactions are performed with periodic boundary conditions that bypass the problem of injecting and expelling ions and water molecules into and out of the computational domain. As stressed, periodic boundary conditions involve several limitations that are particularly serious for ion channel simulations. For example, electrostatic boundary conditions can be applied by acting on the electric field rather than on the electrostatic potential (which is unphysically discontinuous, when not null, at the periodic boundary). It does not constitute a problem from a theoretical standpoint, but it implies a representation of the simulated system that differs from reality, where potentials are applied across the system by using reversible electrodes. These limitations can be addressed by solving Poisson's equation in real space[11] and by devising an appropriate injection/ejection mechanism that mimics the effects of distant electrodes on the small domain being simulated.[164]

Calculating the short-range component of the force field is crucial for an accurate particle-based simulation of ion channels. It is not possible to fully understand the permeation mechanism, especially in narrow channels, without a correct representation of the short-range interactions between the ions and the protein. The need for an accurate representation of short-range interactions is evident in KcsA, where (1) the solvation state of the ions changes during their transit through the channel, and (2) the narrow part of the channel (the selectivity filter) that is lined by backbone carbonyl oxygens, is being traversed by a single line of ions alternating with water molecules. The effective process of selecting and moving ions through such a narrow lumen is the result of competing microscopic interactions[138,155] that must be accounted for precisely. The main problem is related to the force field parameters used to

compute the short-range components, as, for example in the Lennard–Jones potential or in the inverse power relations, as well as the other contributions like bond lengthening, angular deformations, and bond rotational barriers, which are included in the potential energy functions used to describe the molecules under study. Force field parametrization is usually performed in a way that accounts for the average effects of the atomic polarization field, and it involves a redistribution of the electric charge. In other words, the simulated particles (ions and atoms or groups of atoms in the protein and in the membrane) are assigned *effective* properties such as charge, size, and hardness, which normally depend on their position within the system. These force field parameters[165] are optimized in such a way that the simulation reproduces some desired bulk properties of the solution. The idea of the parameterization is to include the effects of the true many-body polar interactions in a simple pair-wise additive fashion, so that the many-body effects can be embedded implicitly in the equations for the short-range force. This approach is efficient from a computational viewpoint, but it involves a few problems. First, the parametrization is not unique.[138] Second, the effects of the real polarization fields are assumed to be fixed rather than consistently evolving with the charge distribution. Finally, the effects of polarization on the molecular flexibility are necessarily neglected. It is safe to assume that in narrow pores, the polarization field plays some role in the structural properties of the protein, and it plays a crucial role in the ion–water and ion–protein interactions. One can try to address the problem by treating the polarization field macroscopically, i.e., by computing an effective position-dependent dielectric tensor at equilibrium,[143] and then use this dielectric tensor in Brownian dynamics or in an appropriately modified parametrization of molecular dynamics. This approach, however, neglects the transient dynamics of the polarization fields that may assist permeation and selectivity in ultra-narrow channels.

The induced point dipole (PD) model and the fluctuating charge (FQ) model are two approaches used to include polarization explicitly in a self-consistent fashion in molecular dynamics simulations.[166,167] Both methods define the total interaction potential as a sum of pair-wise interactions (of all fixed and mobile charges) and an additional polarization term based on the induced electrostatic moments. Within the PD approach, the polarization term is included by ''inducing'' a PD at appropriate charge sites. This dipole depends on all other charges and all other dipoles in the system. Therefore, the total dipole distribution must be computed as a collective (i.e. many-body) property of the system. It can be achieved by an iterative procedure that minimizes the polarization energy,[166] or by treating each dipole as a dynamic variable governed by a set of equations of motion. The polarizable-SPC (PSPC) model is based on this extended Lagrangian formalism, as an example.

With the FQ approach, the polarization term can also be included in the model by changing the amplitude of the charges in response to the electric field. Here, a Lagrangian method is also applied to solve the set of equations

of motion describing the charge dynamics. In this case, the equations are derived for the fluctuating charge system using an electronegativity equalization scheme.[166] In terms of computational resources, the FQ approach is only slightly more demanding than are the nonpolarized charge models, making it attractive. The polarized version of the TIP4P model (TIP4P-FQ) produces RDFs that are in excellent agreement with experimental values.[166]

Molecular dynamics simulations of ionic motion in hydrated ion channels have been performed for more than two decades.[15,168,169] Much work has been done to include explicit polarizability in molecular dynamics simulations[166,167,170–172] and to obtain models of water[146,147,173–175] that can account precisely for the local solvation properties of ions and ion channels.[16,148,176] A more detailed discussion of these approaches cannot be included in this chapter for reasons of space, so the reader is referred to the references indicated above.

FLUX-BASED SIMULATION

It should be clear by now that a microscopic representation of the system components can provide invaluable information for the molecular modeler that relates the structure of ion channels to their function. A major problem with such microscopic, atomistic, particle-based approaches is the inability to perform large-scale simulation in time and space. Even optimistic guesses about the evolution of computer hardware and software place the time frame for modeling all components of a realistically large system, of, say, a few cubic microns and for a few milliseconds, to be decades from now.

Because we want to predict and explain the complex physiological behavior of ion channels on a large scale, one can argue that all information obtained with particle-based approaches is not actually needed. The same issue originated more than a decade ago when algorithms were ranked for their ability to simulate semiconductor devices. State-of-the-art simulation of an individual transistor still takes hours of CPU time for picoseconds of simulated time, whereas key mechanisms like the trap-assisted recombination of charge carriers in the floating body of a silicon on insulator MOSFET have characteristic times of the order of hundreds of microseconds. Furthermore, even if the full characterization of an individual device could be achieved in a reasonable time, what about simulating a whole circuit where thousands of such devices are nonlinearly coupled? Algorithmic efficiency together with faster computing machines are not realistic solutions to such massive and complex problems.

It is our opinion (and probably the most important message that we try to convey in this chapter) that any approach for simulating complex, many-body systems must be based on a hierarchy of consistently related models (see the next section). Each model employed must be validated individually

by comparison (direct or indirect) with experimental data, and the range of validity for each model must be defined as clearly as possible. Knowing in which cases we can safely apply a theory then makes that theory practical.

With this background we devote this section to approaches based on continuum ionic charge distributions rather than limiting this tutorial to the discussion of particle-based simulation methods. These electrodiffusive approaches are called here flux-based because they model the flux of charges, i.e., the current densities, flowing through the system. Electrostatic and van der Waals interactions are accounted for implicitly by a mean field approach where their effects are included as averages over the many instantaneous configurations that the particle-based approaches would otherwise model individually. The main assumption made with the flux-based approaches is therefore that the ion channel behavior can be explained by their mean structural properties rather than by the instantaneous microscopic dynamics of the system.

Nernst–Planck Equation

One continuum model for electrodiffusion of ions between regions of different concentration is based on the combination of Fick's law[177] that describes the diffusion of ions along a concentration gradient and Kohlrausch's law that describes the drift of ions along a potential gradient. Nernst and Planck combined these two laws to obtain the electrodiffusive equation, now known as the Nernst-Planck equation, and which can be written in the Stratonovich form[178,179] as

$$\frac{d}{d\vec{r}}\left\{D(\vec{r})\left[\nabla c(\vec{r}) + \frac{q}{k_B T}\nabla\Phi(\vec{r})c(\vec{r})\right]\right\} = 0 \qquad [53]$$

Here D is the diffusion coefficient, c is the ionic concentration, and Φ is the electrostatic potential caused by the charges within the system and the external boundary conditions. The absolute temperature of the solution is T, whereas k_B is Boltzmann's constant and q is the ionic charge. Integrating Eq. [53] once gives

$$D(\vec{r})\left[\nabla c(\vec{r}) - \frac{1}{k_B T}\vec{F}(\vec{r})c(\vec{r})\right] = -\vec{J} \qquad [54]$$

where $\vec{J}$ is the constant steady-state current density vector and $\vec{F}(\vec{r})$ is the force as calculated through the gradient of the electrostatic potential. Both the concentration c and the force field $\vec{F}$ are space-dependent unknowns in the problem, so two additional equations are coupled to Eq. [54] to obtain $\vec{J}$: the continuity equation

$$\frac{1}{q}\nabla\cdot\vec{J} = -\frac{\partial c}{\partial t} + G \qquad [55]$$

and Poisson's equation (Eq.[18]). The quantity G in Eq.[55] represents the mechanisms of charge generation and recombination occurring within the system. In the following discussion, it is assumed to be null for the sake of simplicity. For a one-dimensional domain starting at $x = 0$ and ending at $x = L$, the solution to Eq. [54] can be written as[180]

$$\vec{J} = \frac{c(0)e^{\Phi(0)/k_BT} - c(L)e^{\Phi(L)/k_BT}}{\int_0^L e^{\Phi(x)/k_BT}\frac{dx}{D(x)}} \qquad [56]$$

The set of three coupled equations (Eq.[18], Eq. [54], and Eq.[55]) is then solved numerically with an iterative procedure that will be discussed in a subsequent section.

The remainder of this section is devoted to the derivation of Eq.[54]. Besides the mathematics we also define the range of applicability of simulations based on the Nernst–Planck equation. The starting point for deriving the Nernst–Planck equation is Langevin's equation (Eq. [45]). A solution of this stochastic differential equation can be obtained by finding the probability that the solution in phase space is $\vec{r}, \vec{v}$ at time t, starting from an initial condition $\vec{r}_0, \vec{v}_0$ at time $t = 0$. This probability is described by the probability density function $p(\vec{r}, \vec{v}, t)$. The basic idea is to find the phase-space probability density function that is a solution to the appropriate partial differential equation, rather than to track the individual Brownian trajectories in phase space. This last point is important, because it defines the difference between particle-based and flux-based simulation strategies.

The derivation of a differential equation for $p(\vec{r}, \vec{v}, t)$ is performed by first defining the diffusion process as an independent Markov process to write a Chapman–Kolmogorov equation in phase space:

$$p(\vec{r} + \vec{v}\Delta t, \vec{v}, t + \Delta t) = \int p(\vec{r}, \vec{v} - \Delta\vec{v}, \Delta t)\Psi(\vec{r}, \vec{v} - \Delta\vec{v}; \Delta\vec{v})d(\Delta\vec{v}) \qquad [57]$$

In Eq. [57], Δt is a time interval chosen to satisfy two criteria: (1) During Δt the position and velocity do not change appreciably, and (2) the stochastic term in Langevin's equation must undergo many fluctuations. Equation [57] states the Markovian nature of $p(\vec{r}, \vec{v}, t)$.

Applying a Taylor expansion to each of the individual terms in Eq. [57] results in the generalization of the Fokker-Planck equation[181] for the evolution of $p(\vec{r}, \vec{v}, t)$ in phase-space:

$$\frac{\partial p}{\partial t} + \vec{v} \cdot \nabla_r p + \frac{\vec{F}}{m} \cdot \nabla_v p = \gamma \nabla_v \cdot (p\vec{v}) + \gamma \frac{k_B T}{m} \nabla_v^2 p \qquad [58]$$

It should be noted that the left-hand side of Eq. [58] is identical to that of the reduced Liouville equation.[181] Indeed, several theories have been developed that obtain Eq. [58] from the reduced Liouville equation.[181,182] Following the standard Smoluchowski expansion[178] of the full time-dependent Fokker–Planck equation, it can be shown that, for large γ, the following model is obtained for the probability density at steady state:

$$\frac{d}{d\vec{r}}\left\{\frac{1}{\gamma}\left[\frac{k_B T}{m}\nabla p(\vec{r}) + \frac{q}{m}\nabla\Phi(\vec{r})p(\vec{r})\right]\right\} = 0 \qquad [59]$$

Note that the dependence of p on the velocity has been dropped because of the overdamping hypothesis (i.e. large γ), and $p(\vec{r}) \equiv \lim_{t\to\infty} p(\vec{r}, t)$, which is a consequence of the steady state hypothesis. Equation [59] is clearly written in the Stratonovich form (see Eq. [53]).

The probability distribution functions in Eq. [59] applied to the trajectories of particles flowing into and out of a system provides a justification for using the Nernst–Planck equation (Eq. [54]): The net ionic directional fluxes can be expressed in terms of differences between the probability fluxes, normalized to the concentration at the sides of the region of interest.[180] That ionic fluxes and differences in probability fluxes are related thus supplies a connection between the solution of the Nernst–Planck equation (Eq. [54]) and the Smoluchowski equation (Eq. [59]), and it provides a direct justification for using Eq. [54] for the study of ions subjected to Brownian dynamics in solution.

Continuing along these lines, we also observe that the Liouville equation is used to obtain the Boltzmann transport equation derived initially within the kinetic theory of gases:

$$\frac{\partial f}{\partial t} + \vec{v}\cdot\nabla_r f + \frac{\vec{F}}{m}\cdot\nabla_v f = \left(\frac{\partial f}{\partial t}\right)_{coll} \qquad [60]$$

where $f(\vec{r}, \vec{v}, t)$ is the phase-space point density, or distribution function, of the particles. The right-hand term of Eq. [60] represents the time rate of change of $f(\vec{r}, \vec{v}, t)$ from collisions that particles undergo within the system.

We now conclude with a derivation of the basic transport equations starting from the Boltzmann equation rather than from the Fokker–Planck equation. We already noted that both the Fokker–Planck and the Boltzmann equations are related to the Liouville equation and that our goal is to obtain equations for the charge distribution and the current density (Eqs. [55] and [54]) using an appropriate representation of the collisional term in the left-hand side of Eq. [60]). The method described here is the well-known method of moments. It consists of multiplying the Boltzmann equation by a power of the velocity, and by integrating over the velocity. For the moment of order zero

(i.e., the zeroth power of the velocity), one uses a constant, which, in this case is the elemental charge q, to obtain

$$\frac{1}{q}\nabla \cdot (qc\vec{v}) = -\frac{\partial c}{\partial t} + G \qquad [61]$$

where G is the integral of the collisional term:

$$G = -\int \left(\frac{df}{dt}\right)_{coll} d\vec{v} \qquad [62]$$

Based on the assumption that collisions change the ionic velocity but not their position, G is simply reduced to a charge generation-recombination term that will be neglected in processes involving transport. By recalling that $\vec{J} = qc\vec{v}$, one realizes that Eq. [61] is the continuity equation previously written (see Eq. [55]).

The moment of order one is obtained by multiplying the Boltzmann equation by $\vec{v}$ and integrating over the velocity space. The result is given by

$$\frac{\partial}{\partial t}(c\vec{v}) + \vec{v}\nabla_r \cdot (c\vec{v}) + (c\vec{v} \cdot \nabla_r)\vec{v} + \frac{1}{m}\nabla_r \cdot (ck_B\hat{T}) + \frac{\vec{F}}{m}c = \left(\frac{\partial \vec{v}}{\partial t}\right)_{coll} \qquad [63]$$

where $\hat{T}$ is the temperature tensor. Equation [63] is simplified considerably by assuming that the concentration c is not a function of time (steady-state assumption). It is accomplished by neglecting the convection term $\vec{v}\nabla \cdot (c\vec{v})$, by representing the tensor $\hat{T}$ with the scalar T, and finally, by assuming that the evolution of the velocity is a sequence of stationary states. Furthermore, assuming that the fluctuations of the velocity generate small deviations from equilibrium, one can apply a relaxation time approximation[183] to the collisional term

$$\left(\frac{\partial \vec{v}}{\partial t}\right)_{coll} \approx \vec{v}\gamma \qquad [64]$$

These assumptions allow us to write Eq. [63] as

$$D(\vec{r})\left[\nabla c(\vec{r}) - \frac{1}{k_B T}\vec{F}(\vec{r})c(\vec{r})\right] = -\vec{J} \qquad [65]$$

which is the Nernst–Planck equation, where the diffusion coefficient is expressed as $D = \frac{k_B T}{\gamma}$.

Because Boltzmann's equation is a conservation relation in phase space, its moments represent conservation laws in position space $\vec{r}$. In particular, the moment of order zero, Eq. [61], is the charge conservation law, whereas Eq. [65] represents current conservation. The next moment is obtained by multiplying Eq. [60] by $mv^2/2$ and integrating over the velocity space. The resulting equation is an energy conservation relation that accounts for heat flow within the system. Inspired by the literature on semiconductor modeling and simulation,[184] Chen et al. solved the system for the first three moments of Boltzmann's equation within the ion channel framework,[185] proposing the inclusion of kinetic energy exchange between the different components of the system.

The PNP Method

This section describes the numerical techniques used for solving the set of differential equations that model the electrodiffusion of ions in solution. The method has historically been called the Poisson–Nernst–Planck (PNP) method because it is based on the coupling of the Poisson equation with the Nernst–Planck equation. The basic equations used in the PNP method include the Poisson equation (Eq. [18]), the charge continuity equation (Eq. [55]), and the current density of the Nernst–Planck equation (Eq. [54]).

Poisson's equation is usually simplified by assuming the dielectric constant to be stepwise constant in the position space. It should be noted that this approximation does not preclude the possibility of having dielectric interfaces within the computational domain; what is assumed here is that the dielectric constant changes abruptly at the interface of different materials. This assumption is completely natural when Poisson's equation is solved on a discrete grid by a finite differences scheme.

Equations [18], [54], and [55] constitute a system of three equations with three unknowns, and this system is solved numerically on one-, two- or three-dimensional domains. For the sake of simplicity, we will discuss the one-dimensional case (the equations are easily extended to three-dimensional). Although finite element methods have been used extensively for the solution of Eqs. [18], and [55] in solid state electronics, flux-based approaches for the simulation of ion channels rely primarily on finite difference schemes.

The system of Eqs. [18], [54], and [55] is usually solved iteratively, with each iteration defined by the successive solution of the three equations. An initial guess is first supplied for the force field $\vec{F}$ in Eq. [54], which is then solved on a discrete grid to provide the components of the current density $\vec{J}$. The divergence of $\vec{J}$ is then computed with the steady-state continuity equation (Eq. [55]) to obtain the charge distribution that, in turn, is used in the forcing function of Poisson's equation. From the gradient of the computed potential, one derives a new (better) approximation to the force $\vec{F}$ that is used to start

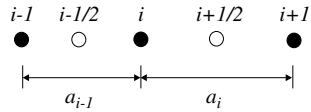


Figure 14 Discretization scheme used for the solution of the PNP equations. Values of the current density are computed at the points designated by empty circles; the potential and charge density values are computed at points corresponding to the filled circles.

another iteration. The iterative process is repeated until the difference between the results of two successive iterations reaches a predefined threshold value. This process ensures self-consistency among the spatial distributions of the charge, current, and potential.

The numerical method for solving of the PNP system in one dimension is normally based on the discretization scheme in Figure 14. A grid is initially defined on which the values of potential and charge distributions are computed (filled circles in Figure 14), whereas the components of the current density vector are computed on points located halfway between those grid-points (empty circles). Because methods for solving Poisson's equation were already discussed, the remaining part of this discussion will focus on the solution of the continuity equation. For a discretization scheme as in Figure 14, one can write a first-order finite difference equation for the continuity equation:

$$\frac{\partial c}{\partial t} = \frac{1}{q}\nabla \cdot \vec{J} = \frac{1}{q}\left[\frac{J_{i+1/2} - J_{i-1/2}}{\frac{a_i + a_{i-1}}{2}}\right] = 0 \qquad [66]$$

where $J_{i+1/2} = J_x(x_{i+1/2})$, a represents the mesh spacing, and the system is assumed to be in steady state. The values of the current can be obtained by finite difference equations obtained from Eq. [54]:

$$J_{i+1/2} = D_{i+1/2}\frac{c_{i+1} - c_i}{a_i} - c_{i+1/2}\frac{D_{i+1/2}}{k_B T}F_{i+1/2} \qquad [67a]$$

$$J_{i-1/2} = D_{i-1/2}\frac{c_i - c_{i-1}}{a_i - 1} - c_{i-1/2}\frac{D_{i-1/2}}{k_B T}F_{i-1/2} \qquad [67b]$$

where the values of any function at the midpoint locations (empty circles in Figure 14) are obtained through linear interpolation. The currents from Eqs. [67a] and [67b] can then be used in Eq. [66] to obtain a difference equation that expresses the concentration c as a function of the force F. In two- and three-dimensional domains, the difference equations are normally solved by using a standard iterative method.

This solution scheme for the PNP method is attractive for its simplicity, but it leads to substantial errors in those regions where large concentration gradients exist. Within this approach, the lack of robustness is traced to the assumption that the ion concentration varies linearly between adjacent grid cells (see Eq. [67]). The discretization of the gradient operator in the current density equation results in negative values for the concentration when the difference of potential between adjacent cells exceeds $2k_BT/q$ V.[186] An effective solution to this problem has been suggested by Sharfetter and Gummel,[187,188] who demonstrated that the discretization errors can be substantially reduced by including a nonlinear exponential variation of ion concentration between grid points. In this case, Eqs. [67a] and [67b] are rewritten as

$$J_{i+1/2} = -qF_{i+1/2}\left[\frac{\frac{D_{i+1/2}}{k_BT}c_{i+1}}{1-\exp[\frac{F_{1+1/2}}{q}a_i]} + \frac{\frac{D_{i+1/2}}{k_BT}c_i}{1-\exp[\frac{-F_{1+1/2}}{q}a_i]}\right] \quad [68a]$$

$$J_{i-1/2} = -qF_{i-1/2}\left[\frac{\frac{D_{i-1/2}}{k_BT}c_i}{1-\exp\left[\frac{F_{1-1/2}}{q}a_{i-1}\right]} + \frac{\frac{D_{i-1/2}}{k_BT}c_{i-1}}{1-\exp\left[\frac{-F_{1-1/2}}{q}a_{i-1}\right]}\right] \quad [68b]$$

Again, the current from Eqs. [68a] and [68b] can be used to obtain a difference scheme for the concentration via the continuity equation. The method of Sharfetter and Gummel is slightly slower than the linearized approach, but it is more accurate and is remarkably more robust in the presence of high concentration gradients. Furthermore, it produces positive-definite matrices and hence can be implemented by using overrelaxation techniques,[119] which have a relatively fast rate of convergence. Multigrid methods[125] have also been used successfully for solving PNP-like equations (see Molenaar[189] and references therein). Finally, it should be noted that the approach suggested by Eisenberg et al.[180] (which is also based on the iterative solution of the continuity equation, the analytical solution of the Nernst–Planck equation (Eq. [56]), and Poisson's equation (Eq. [18])) is mathematically equivalent to the Gummel iteration once it has been discretized on a finite difference grid.

The evolution of the numerical approaches used for solving the PNP equations has paralleled the evolution of computing hardware. The numerical solution to the PNP equations evolved over the time period of a couple of decades beginning with the simulation of extremely simplified structures[84,190] to fully three-dimensional models,[22,191,192] and with the implementation of sophisticated variants of the algorithmic schemes to increase robustness and performance.[20] Even finite element tetrahedral discretization schemes have been employed successfully to selectively increase the resolution in regions inside the channels.[21] An important aspect of the numerical procedures described is the need for full self-consistency between the force field and the charge distribution in space. It is obtained by coupling a Poisson solver to the Nernst–Planck solver[1,193] within the iteration scheme described.

The PNP approach, together with the Poisson–Boltzmann[194,195] method, belong to the family of continuum theories of electrolytes[196] that are based on the mean field approximation. Because the PNP approach is based on continuous fluxes rather than on individual trajectories, average concentrations are employed and the ions are assumed to move in average electric fields.[1] Consequently, a key role is played by *macroscopic* parameters such as the diffusion coefficient and the dielectric constant. Recall that the term "macroscopic" is being used here to represent the *collective* behavior of a large group of microscopic components of the system, i.e., atoms within molecules and ions in solution. The PNP theory has been developed as a model for large systems, e.g., those with feature-sizes larger than the Debye length, and its applicability in modeling ionic permeation within channels only a few angstroms across has been questioned.[197,198] Similar concerns have been expressed with respect to the Poisson–Boltzmann method.[199] It should be noted, however, that the relevant Debye length in either method is that within the channel (or active site) and not that within the bulk. The concentration of ions is typically 10–50 times higher inside a channel than in bulk, and consequently, the Debye length is extremely small there.

From a physical viewpoint, the use of a fixed diffusion coefficient corresponds to assuming that the ionic energy relaxation time is independent of the local electric field. The same approximation is applicable to the friction coefficient γ in Langevin's equation. Generally speaking, ion channels are not ohmic machines, at least during the transient conditions typical of gating. Because of the steady-state assumption, the PNP method is not suitable for the study of fast transients or situations in which the ionic energy is different from the energy of the surrounding system. Nonetheless, because it can supply valuable information for the study of the steady-state ohmic regime, assuming a constant diffusivity has been a popular choice by scientists studying ion channels. Alternatively, and similarly to what is done for the simulation of semiconductor devices, a space-dependent diffusivity for a specific channel configuration can be obtained with particle-based simulation (usually molecular dynamics) and used in the PNP equations.[160]

Solving the PNP equations is more complicated when including the effects of polarization. Defining a dielectric "constant" within a small channel presents huge conceptual difficulties. From a microscopic viewpoint, the dynamics of one ion (or more) within the channel is being described as a purely many-body problem, meaning that the presence of even one ion modifies significantly the polarization field felt by the ion and, possibly, even the structural conformation of the channel protein. It is reasonable to conclude that the polarization plays a role in the conduction properties (including selectivity) of very narrow channels and possibly even wider channels. The interaction of ions with dielectric surfaces is definitely a local phenomenon related to the particle character of the ions. However, continuum theories represent ions as a smooth charge distribution rather than as point-like charges, and

modeling effects of the dielectric interfaces on such distributed charge is thus particularly arduous. A typical example is the problem of "overshielding" shown by PNP simulations of narrow channels. The continuous nature of PNP results in the formation of a spurious countercharge in the channel that is already populated by a given ionic species. This flow of countercharge represents counterions that would not normally enter the channel because of their interaction with the image charges generated at the channel dielectric surface. This spurious and nonrealistic effect, in contrast, is not produced by particle-based Brownian dynamics simulations because the finite size of ions is included in Brownian dynamics. The spurious countercharge modifies the electrostatic landscape, and consequently, a remarkable discrepancy is found in the ionic concentrations when comparing results obtained with Brownian dynamics.[198] Several adjustments to the PNP theory have been proposed to alleviate this problem, either via the inclusion of a term that accounts for the induced image charges as a surface charge in Poisson's equation[200] or, by correcting the free energy of the system with a potential of mean force[181] obtained from molecular dynamics simulations.[160] These extended theories offer better agreement with results from particle-based approaches,[201] at least for single-channel occupancy and in very narrow channels.

In conclusion, flux-based approaches are appealing because of their low computational costs and because of their ability to predict quantities that are directly observable, such as currents flowing through open channels. Their utility for the study of small channels has been questioned, especially because of the argued inability to account for molecular flexibility. Therefore, much theoretical work is needed to extend and generalize the flux-based simulation approaches to better account for more realistic configurations, keeping in mind their basic limitations. Several researchers rightly stressed the need for validating flux-based simulation with microscopic particle-based models. Analogously, particle-based models must also be validated on the largest scales for which they are used, in the hierarchy of models (see below).

HIERARCHICAL SIMULATION SCHEMES

Several approaches for the simulation of ionic charge transport in protein channels have been presented in the previous sections. It should be clear from this discussion that none of the mentioned methods can supply a complete and self-contained description of the full functionality of ion channels starting from purely structural information. For this reason, methods based on a hierarchy of simulative approaches,[202,203] rather than on a specific method, are becoming more and more popular.

The concept of using atomistic molecular dynamics simulations for extracting parameters to be used in less-precise but faster Brownian dynamics simulators, or electrodiffusive solvers, has been discussed. This methodology

can be applied extensively, and it can entail molecular mechanics techniques for the full preparation of the protein structure, continuum techniques for its electrostatic characterization, and molecular dynamics for the extraction of diffusion or energy profiles for use in Brownian dynamics.[204] This ''sequential'' approach has been used with excellent results in other fields, and it is well established in computational biophysics.[205] A further step in the direction of hierarchical modeling is to use different approaches simultaneously and to analyze the (sometimes different) results by keeping in mind strengths and weaknesses of the simulative methodologies. This ''parallel'' or comparative strategy has a certain degree of subjectivity that can be minimized if a rigorous attitude is adopted by the modeler throughout the study when interpreting the results[51] and by calibrating the methods.

Furthermore, because of the limitations in the size of the systems that can be simulated with high-resolution, all-atom techniques, the integration of different approaches into the same simulation procedure is necessary in the foreseeable future. For example, using a molecular dynamics simulation engine in a relatively small region of the system under study and including a larger domain where the solvent can be treated implicitly, say with Brownian dynamics, will allow for extending the size of the simulated system and to reduce artifacts originating from close boundaries. Of course, the correct treatment of the interface that ''bridges'' two regions simulated with different computational and philosophical approaches is not trivial and solving such problems has not yet been accomplished.

FUTURE DIRECTIONS AND CONCLUDING REMARKS

The recent development of high-resolution experimental techniques allows for the structural analysis of protein channels with unprecedented detail. However, the fundamental problem of relating the structure of ion channels to their function is a formidable task. This chapter describes some of the most popular simulation approaches used to model channel systems. Particle-based approaches such as Brownian and molecular dynamics will continue to play a major role in the study of protein channels and in validating the results obtained with the extremely fast continuum models. Research in the area of atomistic simulations will focus mainly on the force-field schemes used in the ionic dynamics simulation engines. In particular, polar interactions between the various components of the system need to be computed with algorithms that are more accurate than those currently used. The effects of the local polarization fields need to be accounted for explicitly and, at the same time, efficiently. Continuum models will remain attractive for their efficiency in depicting the electrostatic landscape of protein channels. Both Poisson–Boltzmann and Poisson–Nernst–Plank solvers will continue to be used to

extract qualitative information about the macroscopic behavior of ion channels. Their quantitative predictions will become better valued once the range of applicability of the theory is validated by particle-based approaches. These improvements will help scientists address how the functional properties of ion channels depend on the instantaneous structural fluctuations and to what extent a mean conformational characteristic is sufficient to describe these amazingly complex systems.

The idea of integrating different approaches into a hierarchical simulation strategy is promising. It can be accomplished either through a "sequential" approach, in which the results obtained with one method are used to calibrate a faster but less accurate one, or through a "concurrent" technique, in which several simulation tools are integrated or "bridged" within the same algorithm in a way that provides different levels of accuracy in different regions of the computational domain.

REFERENCES

1. B. Hille, *Ionic Channels of Excitable Membranes*, Third Edition, Sinauer, Sunderland, Massachusetts, 2001.
2. E. Neher and B. Sackmann, *Nature*, **260**, 799 (1976). Single-Channel Currents Recorded from Membrane of Denervated Frog Muscle Fibers.
3. B. Sakmann and E. Neher, Eds., *Single-Channel Recording*, Second Edition, Kluwer Academic, New York, 1995.
4. F. M. Ashcroft, *Ion Channels and Disease*, Academic Press, San Diego, California, 2000.
5. L.-Q. Gu, O. Braha, S. Conlan, S. Cheley, and H. Bayley, *Nature*, **398**, 686 (1999). Stochastic Sensing of Organic Analytes by a Pore-forming Protein Containing a Molecular Adapter.
6. H. Bayley and P. S. Cremer, *Nature*, **413**, 226 (2001). Stochastic Sensors Inspired by Biology.
7. M. Goryll, S. Wilk, G. M. Laws, T. Thornton, S. Goodnick, M. Saraniti, J. Tang, and R. S. Eisenberg, *Superlattices Microstruct.*, **34**(3–6), 451–457 (2003). Silicon-based Ion Channel Sensor.
8. S. Sze, *Physics of Semiconductor Devices*, Second Edition, John Wiley & Sons, New York, 1981.
9. Semiconductor Industry Association, in International Technology Roadmap for Semiconductors, International SEMATECH, Austin, Texas, 1999, pp. 83–103. Process Integration, Devices, and Structures.
10. E. R. Davidson, in *Reviews in Computational Chemistry*, Vol. 6, K. B. Lipkowitz and D. B. Boyd, Eds., VHC Publishers, New York, 1990, pp. 373–382. Perspectives on *Ab Initio* Calculations.
11. H. J. C. Berendsen, in *Computational Molecular Dynamics: Challenges, Methods, Ideas*, P. Deuflhard, Ed., Springer-Verlag, 1999, pp. 3–36.
12. P. Gibbon and G. Sutmann, in *Quantum Simulations of Many-Body Systems: From Theory to Algorithms, Lecture Notes*, Vol. 10, J. Grotendorst, D. Marx, and A. Muramatsu, Eds., John von Neumann Institute for Computing, Jülich, Germany, 2002, pp. 467–506.
13. D. E. Elmore and D. A. Dougherty, *Biophys. J.*, **85**(3), 1512 (2003). Investigating Lipid Composition Effects on the Mechanosensitive Channel of Large Conductance (MscL) Using Molecular Dynamics Simulations.

14. B. L. de Groot, D. P. Tieleman, P. Pohl, and H. Grubmuller, *Biophys. J.*, **82**(6), 2934 (2002). Water Permeation through Gramicidin A: Desformylation and the Double Helix: A Molecular Dynamics Study.
15. W. K. Lee and P. C. Jordan, *Biophys. J.*, **46**(6), 805 (1984). Molecular Dynamics Simulation of Cation Motion in Water-Filled Gramicidin like Pores.
16. S. W. Chiu, S. Subramaniam, E. Jakobsson, and J. A. McCammon, *Biophys. J.*, **56**(2), 253 (1989). Water and Polypeptide Conformations in the Gramicidin Channel. A Molecular Dynamics Study.
17. B. Roux and M. Karplus, *Biophys. J.*, **59**(5), 961 (1991). Ion Transport in a Model Gramicidin Channel. Structure and Thermodynamics.
18. J. A. Szule and R. P. Rand, *Biophys. J.*, **85**(3), 1702 (2003). The Effects of Gramicidin on the Structure of Phospholipid Assemblies.
19. T. W. Allen, O. S. Anderson, and B. Roux, *Proc. Natl. Acad. Sci. USA*, **101**(1), 117 (2004). Energetics of Ion Conduction Through the Gramicidin Channel.
20. M. G. Kurnikova, R. D. Coalson, P. Graf, and A. Nitzan, *Biophys. J.*, **76**(2), 642 (1999). A Lattice Relaxation Algorithm for Three-Dimensional Poisson-Nernst-Planck Theory with Application to Ion Transport Through the Gramicidin A Channel.
21. U. Hollerbach, D. P. Chen, D. D. Busath, and B. Eisenberg, *Langmuir*, **16**(13), 5509 (2000). Predicting Function from Structure using the Poisson-Nernst-Plank Equations: Sodium Current in the Gramicidin A Channel.
22. U. Hollerbach, D. P. Chen, and R. S. Eisenberg, *J. Sci. Comput.*, **16**(4), 373 (2001). Two- and Three-Dimensional Poisson-Nernst-Plank Simulations of Current Flow Through Gramicidin A.
23. S. Edwards, B. Corry, S. Kuyucak, and S.-H. Chung, *Biophys. J.*, **83**(3), 1348 (2002). Continuum Electrostatics Fails to Describe Ion Permeation in the Gramicidin Channel.
24. T. W. Allen, T. Bastug, S. Kuyucak, and S.-H. Chung, *Biophys. J.*, **84**(4), 2159 (2003). Gramicidin A Channel as a Test Ground for Molecular Dynamics Force Fields.
25. D. W. Urry, *Proc. Natl. Acad. Sci. USA*, **68**, 672 (1971). The Gramicidin A Transmembrane Channel: A Proposed π_{LD} Helix.
26. R. R. Ketchem, W. Hu, and T. A. Cross, *Science*, **261**, 1457 (1993). High-Resolution Conformation of Gramicidin A in a Lipid Bilayer by Solid-State NMR.
27. L. E. Townsley, W. Tucker, S. Sham, and J. F. Hinton, *Biochemistry*, **40**, 11676 (2001). Structures of Gramicidin A, B and C Incorporated in Sodium Dodecyl Sulfate Micelles.
28. T. W. Allen, O. S. Andersen, and B. Roux, *J. Am. Chem. Soc.*, **125**(32), 9868 (2003). Structure of Gramicidin a in a Lipid Bilayer Environment Determined Using Molecular Dynamics Simulations and Solid-State NMR Data.
29. Y. Chen, A. Tucker, and B. A. Wallace, *J. Mol. Biol.*, **264**(4), 757 (1996). Solution Structure of a Parallel Left-Handed Double-Helical Gramicidin-A Determined by 2D ^{1}H NMR.
30. R. R. Ketchem, W. Hu, and T. A. Cross, *J. Biomol. NMR*, **8**(1), 1 (1996). Macromolecular Structure Elucidation with Solid-State NMR-Derived Orientation Constraints.
31. T. A. Harroun, W. T. Heller, T. M. Weiss, L. Yang, and H. W. Huang, *Biophys. J.*, **76**(6), 3176 (1999). Theoretical Analysis of Hydrophobic Matching and Membrane-Mediated Interactions in Lipid Bilayers Containing Gramicidin.
32. S.-W. Chiu, S. Subramaniam, and E. Jakobsson, *Biophys. J.*, **76**(4), 1929 (1999). Simulation Study of a Gramicidin/Lipid Bilayer System in Excess Water and Lipid. I. Structure of the Molecular Complex.
33. T. A. Harroun, W. T. Heller, T. M. Weiss, L. Yang, and H. W. Huang, *Biophys. J.*, **76**(2), 937 (1999). Experimental Evidence for Hydrophobic Matching and Membrane-Mediated Interactions in Lipid Bilayers Containing Gramicidin.
34. S.-W. Chiu, S. Subramaniam, and E. Jakobsson, *Biophys. J.*, **76**(4), 1939 (1999). Simulation Study of a Gramicidin/Lipid Bilayer System in Excess Water and Lipid. II. Rates and Mechanisms of Water Transport.

35. G. V. Miloshevsky and P. C. Jordan, *Biophys. J.*, **86**(1), 92 (2004). Gating Gramicidin Channels in Lipid Bilayers: Reaction Coordinates and the Mechanism of Dissociation.

36. K. M. Armstrong and S. Cukierman, *Biophys. J.*, **82**(3), 1329 (2002). On the Origin of Closing Flickers in Gramicidin Channels: A New Hypothesis.

37. G. S. Harms, G. Orr, M. Montal, B. D. Thrall, S. D. Colson, and H. P. Lu, *Biophys. J.*, **85**(3), 1826 (2003). Probing Conformational Changes of Gramicidin Ion Channels by Single-Molecule Patch-Clamp Fluorescence Microscopy.

38. D. P. Tieleman, P. C. Bigging, G. R. Smith, and M. S. P. Sansom, *Quart. Rev. Biophys.*, **34**(4), 473 (2001). Simulation Approaches to Ion Channel Strucuture-Function Relationships.

39. W. Humphrey, A. Dalke, and K. Schulten, *J. Mol. Graphics Model.*, **14**(1), 33 (1996). VMD-Visual Molecular Dynamics.

40. D. A. Doyle, J. M. Cabral, R. A. Pfuetzner, A. Kuo, J. M. Gulbis, S. M. Cohen, B. T. Chait, and R. MacKinnon, *Science*, **280**, 69 (1998). The Structure of the Potassium Channel: Molecular Basis of K^+ Conduction and Selectivity.

41. Y. Zhou, J. H. Morales-Cabral, A. Kaufman, and R. MacKinnon, *Nature*, **414**, 43 (2001). Chemistry of Ion Coordination and Hydration Revealed by a K^+ Channel-Fab Complex at 2.0 Å Resolution.

42. Y. Jiang, A. Lee, J. Chen, M. Cadene, B. T. Chait, and R. MacKinnon, *Nature*, **417**, 515 (2002). Crystal Structure and Mechanism of a Calcium-Gated Potassium Channel.

43. G. Yellen, *Nature*, **419**, 35 (2002). The Voltage-Gated Potassium Channels and Their Relatives.

44. Y. Jiang, A. Lee, J. Chen, V. Ruta, M. Cadene, B. T. Chait, and R. MacKinnon, *Nature*, **423**, 33 (2003). X-ray Structure of a Voltage-Dependent K^+ Channel.

45. E. Perozo, D. M. Cortes, and L. G. Cuello, *Science*, **285**, 73 (1999). Structural Rearrangements Undelying K^+-Channel Activation Gating.

46. Y. Jiang, V. Ruta, J. Chen, A. Lee, and R. MacKinnon, *Nature*, **423**, 42 (2003). The Principle of Gating Charge Movement in a Voltage-Dependent K^+ Channel.

47. C. A. Ahern and R. Horn, *J. Gen. Physiol.*, **123**, 205 (2004). Specificity of Charge-Carrying Residues in the Voltage Sensor of Potassium Channels.

48. B. Roux and R. MacKinnon, *Science*, **285**, 100 (1999). The Cavity and Pore Helices in the KcsA K^+ Channel: Electrostatic Stabilization of Monovalent Cations.

49. E. M. Nestorovich, T. K. Rostovtseva, and S. M. Bezrukov, *Biophys. J.*, **85**(6), 3718 (2003). Residue Ionization and Ion Transport Through OmpF Channels.

50. S. Varma and E. Jakobsson, *Biophys. J.*, **86**(2), 690 (2004). Ionization States of Residues in OmpF and Mutants: Effects of Dielectric Constant and Interactions Between Residues.

51. W. Im and B. Roux, *J. Mol. Biol.*, **322**(4), 851 (2002). Ion Permeation and Selectivity of OmpF Porin: A Theoretical Study Based on Molecular Dynamics, Brownian Dynamics, and Continuum Electrodiffusion Theory.

52. W. Im and B. Roux, *J. Mol. Biol.*, **319**(5), 1177 (2002). Ions and Counterions in a Biological Channel: A Molecular Dynamics Simulation of OmpF Porin from Escherichia Coli in an Explicit Membrane with 1 M KCl Aqueous Salt Solution.

53. A. Karshikoff, V. Spassov, S. W. Cowan, R. Ladenstein, and T. Schirmer, *J. Mol. Biol.*, **240**, 372 (1994). Electrostatic Properties of Two Porin Channels from Escherichia Coli.

54. A. Philippsen, W. Im, A. Engel, T. Schirmer, B. Roux, and D. J. Muller, *Biophys. J.*, **82**(3), 1667 (2002). Imaging the Electrostatic Potential of Transmembrane Channels: Atomic Probe Microscopy of OmpF Porin.

55. B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Essential Cell Biology*, Garland Publishing, New York, 1998.

56. K. V. Damodaran and K. M. Merz, Jr., in *Reviews in Computational Chemistry*, Vol. 5, K. B. Lipkowitz and D. B. Boyd, Eds., VHC Publishers, New York, 1994, pp. 269–298. Computer Simulation of Lipid Systems.

57. J. M. Berg, J. L. Tymoczko, and L. Stryer, *Biochemistry*, Fifth Edition, W. H. Freeman, New York, 2001.

58. D. P. Tieleman, M. S. P. Sansom, and H. J. C. Berendsen, *Biophys. J.*, **76**(1), 40 (1999). Alamethicin Helices in a Bilayer and in Solution: Molecular Dynamics Simulations.

59. J. F. Nagle and S. Trsitram-Nagle, *Curr. Opin. Struct. Biol.*, **10**(4), 474 (2000). Lipid Bilayer Structure.

60. H. L. Scott, *Curr. Opin. Struct. Biol.*, **12**, 495 (2002). Modeling the Lipid Component of Membranes.

61. U. Essmann and M. L. Berkowitz, *Biophys. J.*, **76**(4), 2081 (1999). Dynamical Properties of Phospholid Bilayers from Computer Simulations.

62. E. Lindahl and O. Edholm, *Biophys. J.*, **79**(1), 426 (2000). Mesoscopic Undulations and Thickness Fluctuations in Lipid Bilayers from Molecular Dynamics Simulations.

63. I. P. Sugar, T. E. Thompson, and R. L. Biltonen, *Biophys. J.*, **76**(4), 2099 (1999). Monte Carlo Simulations of Two-Component Bilayers: DMPC/DSPC Mixtures.

64. A. Smondyrev and M. L. Berkowitz, *Biophys. J.*, **77**, 2075 (1999). Structure of Dipalmitoylphosphatidylcholine/Cholesterol Bilayer at Low and High Cholesterol Concentrations: Molecular Dynamics Simulations.

65. S. W. Chiu, E. Jakobsson, and S. Subramaniam, *Biophys. J.*, **77**, 2462 (1999). Combined Monte Carlo and Molecular Dynamics Simulation of Fully Hydrated Dioleyl and Palmitoyloleyl Phosphatidylcholine Lipid Bilayers.

66. S. E. Feller, *Curr. Opin. Colloid Interface Sci.*, **5**, 217 (2000). Molecular Dynamics Simulations of Lipid Bilayers.

67. E. Jakobsson, *Trends Biochem. Sci.*, **22**, 339 (1997). Computer Simulation Studies of Biological Membranes: Progress, Promise and Pittfalls.

68. G. Hummer, L. R. Pratt, and A. E. Garćia, *J. Phys. Chem. A*, **102**(41), 7885 (1998). Molecular Theories and Simulation of Ions and Polar Molecules in Water.

69. J. Bockris and A. K. N. Reddy, *Modern Electrochemistry*, Second Edition, Vol. I, Ionics, Plenum Press, New York, 1998.

70. S. C. Li, M. Hoyles, S. Kuyucak, and S.-H. Chung, *Biophys. J.*, **74**(1), 37 (1998). Brownian Dynamics Study of Ion Transport in the Vestibule of Membrane Channels.

71. J.-P. Simonin, L. Blum, and P. Turq, *J. Phys. Chem.*, **100**(18), 7704 (1996). Real Ionic Solutions in the Mean Spherical Approximation. 1. Simple Salts in the Primitive Model.

72. J. M. G. Barthel, H. Krienke, and W. Kunz, in *Topics in Physical Chemistry*, Vol. 5, Springer, New York, 1998. Physical Chemistry of Electrolyte Solutions.

73. S. Durand-Vidal, J.-P. Simonin, and P. Turq, *Electrolytes at Interfaces*, Kluwer, Norwell, Massachusetts, 2000.

74. W. Im, M. Feig, and C. L. Brooks, III, *Biophys. J.*, **85**(5), 2900 (2003). An Implicit Membrane Generalized Born Theory for the Study of Structure, Stability, and Interactions of Membrane Proteins.

75. D. P. Tieleman, L. R. Forrest, M. S. P. Sansom, and H. J. C. Berendsen, *Biochemistry*, **37**, 17554 (1998). Lipid Properties and the Orientation of Aromatic Residues in OmpF, Influenza M2, and Alamethicin Systems: Molecular Dynamics Systems.

76. T. Schirmer and P. Phale, *J. Mol. Biol.*, **294**, 1159 (1999). Brownian Dynamics Simulation of Ion Flow Through Porin Channels.

77. T. B. Wolf and B. Roux, *Protein: Struct., Funct. and Genet.*, **24**(1), 92 (1996). Structure, Energetics, and Dynamics of Lipid-Protein Interactions: A Molecular Dynamics Study of Gramicidin A Channel in a DMPC Bilayer.

78. T. P. Lybrand, in *Reviews in Computational Chemistry*, Vol. 1, K. B. Lipkowitz and D. B. Boyd, Eds., VHC Publishers, New York, 1990, pp. 295–320. Computer Simulation of Biomolecular Systems Using Molecular Dynamics and Free Energy Perturbation Methods.

79. D. P. Tieleman and H. J. C. Berendsen, *Biophys. J.*, **74**(6), 2786 (1998). A Molecular Dynamics Study of the Pores Formed by Escherichia coli OmpF Porin in a Fully Hydrated Palmitoyloleoylphosphatidylcholine Bilayer.

80. L. D. Shen, D. Bassolino, and T. Stouch, *Biochemistry*, **73**, 3 (1997). Transmembrane Helix Structure, Dynamics, and Interactions: Multi-Nanosecond Molecular Dynamics Simulations.

81. P. G. Mezey, in *Reviews in Computational Chemistry*, Vol. 1, K. B. Lipkowitz and D. B. Boyd, Eds., VHC Publishers, New York, 1990, pp. 265–294. Molecular Surfaces.

82. J. D. Faraldo-Gomez, G. R. Smith, and M. S. P. Sansom, *Eur. Biophys. J.*, **31**, 217 (2002). Setting Up and Optimisation of Membrane Protein Simulations.

83. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*, Adam Hilger, Bristol, United Kingdom, 1988.

84. D. G. Levitt, *Biophys. J.*, **48**(1), 19 (1985). Strong Electrolyte Continuum Theory Solution for Equilibrium Profiles, Diffusion Limitation, and Conductance in Charged Ion Channels.

85. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford, United Kingdom, 1987.

86. Y. Georgalis, A. M. Kierzek, and W. Sa, *J. Phys. Chem. B*, **104**, 3405 (2000). Cluster Formation in Aqueous Electrolyte Solutions Observed by Dynamic Light Scattering.

87. C. Jacoboni and P. Lugli, *The Monte Carlo Method for Semiconductor Device Equations*, Springer-Verlag, NewYork, 1989.

88. P. C. Jordan, *Biophys. J.*, **39**(2), 157 (1982). Electrostatic Modeling of Ion Pores. I. Energy Barriers and Electric Field Profiles.

89. P. C. Jordan, *Biophys. J.*, **41**(2), 189 (1983). Electrostatic Modeling of Ion Pores. II. Effects Attributable to the Membrane Dipole Potential.

90. M. Cai and P. C. Jordan, *Biophys. J.*, **57**(4), 883 (1990). How Does Vestibule Surface Charge Affect Ion Conduction and Toxin Binding in a Sodium Channel?

91. G. V. Miloshevsky and P. C. Jordan, *Biophys. J.*, **86**(2), 825 (2004). Anion Pathway and Potential Energy Profiles along Curvilinear Bacterial ClC Cl^- Pores: Electrostatic Effects of Charged Residues.

92. W. Nonner, D. P. Chen, and B. Eisenberg, *Biophys. J.*, **74**(5), 2327 (1998). Anomalous Mole Fraction Effect, Electrostatics, and Binding in Ionic Channels.

93. W. Nonner, L. Catacuzzeno, and B. Eisenberg, *Biophys. J.*, **79**(4), 1976 (2000). Binding and Selectivity in L-Type Calcium Channels: A Mean Spherical Approximation.

94. D. Boda, D. D. Busath, D. Henderson, and S. Sokolowski, *J. Phys. Chem. B*, **104**(37), 8903 (2000). Monte Carlo Simulations of the Mechanism for Channel Selectivity: The Competition between Volume Exclusion and Charge Neutrality.

95. D. Boda, D. Gillespie, W. Nonner, D. Henderson, and B. Eisenberg, *Phys. Rev. E*, **69**, 046702 (2004). Computing Induced Charges in Inhomogeneous Dielectric Media: Application in a Monte Carlo Simulation of Complex Ionic Systems.

96. D. Boda, D. Henderson, and D. D. Busath, *J. Phys. Chem. B*, **105**(47), 11574 (2001). Monte Carlo Study of the Effect of Ion and Channel Size on the Selectivity of a Model Calcium Channel.

97. L. Greengard, *Science*, **265**(5174), 909 (1994). Fast Algorithms for Classical Physics.

98. L. Greengard and V. Rokhlin, *J. Comput. Phys.*, **135**, 280 (1997). A Fast Algorithm for Particle Simulations.

99. F. S. Lee and A. Warshel, *J. Chem. Phys.*, **97**(5), 3100 (1992). A Local Field Method for Fast Evaluation of Long-Range Electrostatic Interactions in Molecular Simulations.

100. P. Ewald, *Ann. Phys.*, **64**, 253 (1921). Die Berechnung Optischer und Elektrostatischer Gitterpotentiale.

101. M. Deserno and C. Holm, *J. Chem. Phys.*, **109**(18), 7678 (1998). How to Mesh Up Ewald Sums. I. A Theoretical and Numerical Comparison of Various Particle Mesh Routines.

102. J. D. Jackson, *Classical Electrodynamics*, Second Edition, John Wiley & Sons, New York, 1975.
103. H.-Q. Ding, N. Karasawa, and W. A. Goddard, III, *J. Chem. Phys.*, **97**(6), 4309 (1992). Atomic Level Simulations on a Million Particles: The Cell Mulipole Method for Coulomb and London Nonbonded Interactions.
104. J. Shimada, H. Kaneko, and T. Takada, *J. Comput. Chem.*, **15**(1), 29 (1994). Performance of Fast Mulipole Methods for Calculating Electrostatic Interactions in Biomacromolecular Simulations.
105. C. A. White and M. Head-Gordon, *J. Chem. Phys.*, **101**(8), 6593 (1994). Derivation and Efficient Implementation of the Fast Mulitpole Method.
106. T. Schlick, in *Interdisciplinary Applied Mathematics*, Vol. 21, Springer, New York, 2000. Molecular Modeling and Simulation: An Interdisciplinary Guide.
107. A. W. Appel, *SIAM J. Sci. Stat. Comput.*, **6**(1), 85 (1985). An Efficient Program for Many-Body Problems.
108. E. L. Pollock and J. Glosli, *Comput. Phys. Commun.*, **95**, 93 (1996). Comments on P^3M, FMM, and the Ewald Method for Large Periodic Coulombic Systems.
109. A. R. Leach, *Molecular: Modelling Principles and Applications*, Second Edition, Prentice-Hall, Harlow, United Kingdom, 2001.
110. P. Crozier, R. L. Rowley, D. Henderson, and D. Boda, *Chem. Phys. Lett.*, **325**(5-6), 675 (2000). A Corrected 3D Ewald Calculation of the Low Effective Temperature Properties of the Electrochemical Interface.
111. D. M. Heyes, *J. Chem. Phys.*, **74**(3), 1924 (1981). Electrostatic Potentials and Fields in Infinite Point Charge Lattices.
112. T. A. Darden, D. York, and L. Pedersen, *J. Chem. Phys.*, **98**(12), 10089 (1993). Particle Mesh Ewald: An Nlog(N) Method for Ewald Sums in Large Systems.
113. A. Y. Toukmaji and J. A. Board, *Comput. Phys. Commun.*, **95**, 73 (1996). Ewald Summation Techniques in Perspective: A Survey.
114. N. Karasawa and W. A. Goddard, III, *J. Phys. Chem.*, **93**, 7320 (1989). Acceleration of Convergence for Lattice Sums.
115. G. Rajagopal and R. M. L. J. Needs, *J. Comput. Phys.*, **115**, 399 (1994). An Optimized Ewald Method for Long-Ranged Potentials.
116. U. Essmann, L. Perera, M. L. Berkowitz, T. A. Darden, H. Lee, and L. Pedersen, *J. Chem. Phys.*, **103**(19), 8577 (1995). A Smooth Particle Mesh Ewald Method.
117. C. Pommerell and W. Fichtner, *SIAM J. Sci. Stat. Comput.*, **15**(2), 460 (1994). Memory Aspects and Performance of Iterative Solvers.
118. D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.
119. G. Dahlquist and Å. Björck, *Numerical Methods*, Prentice–Hall, Englewood Cliffs, New Jersey, 1974.
120. P. J. Roache, *Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, New Mexico, 1976.
121. B. A. Carré, *Comput. J.*, **4**(1), 73 (1961). The Determination of the Optimum Accelerating Factor for Successive Over–Relaxation.
122. L. W. Ehrlich, in *Elliptic Problem Solvers*, M. H. Schultz, Ed., Academic Press, New York, 1981, pp. 255–259.
123. L. M. Adams and H. F. Jordan, *SIAM J. Sci. Stat. Comput.*, **7**(2), 490 (1986). Is SOR Color–Blind?
124. S. Vandewalle, in *Parallel Multigrid Waveform Relaxation for Parabolic Problems*, B.G. Teubner, Stuttgart, Germany, 1993. Teubner Skripten zur Numerik.
125. W. Hackbush, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, Germany, 1985.

126. A. Brandt, *Math. Comput.*, **31**(138), 333 (1977). Multi-Level Adaptive Solutions to Boundary-Value Problems.

127. A. Brandt, in *Elliptic Problem Solvers*, M. H. Schultz, Ed., Academic Press, New York, 1981, pp. 39–84. Multigrid Solvers on Parallel Computers.

128. P. S. Ramanathan and H. L. Friedman, *J. Chem. Phys.*, **54**(3), 1086 (1971). Study of a Refined Model for Aqueous 1-1 Electrolytes.

129. R. S. Berry, S. A. Rice, and J. Ross, *Physical Chemistry*, Second Edition, Oxford University Press, Oxford, United Kingdom, 2000.

130. W. Im, S. Seefeld, and B. Roux, *Biophys. J.*, **79**(2), 788 (2000). A Grand Canonical Monte Carlo-Brownian Dynamics Algorithm for Simulating Ion Channels.

131. M. A. Wilson, A. Pohrille, and L. R. Pratt, *J. Chem. Phys.*, **83**(11), 5832 (1985). Molecular Dynamic Test of the Brownian Description of Na^+ Motion in Water.

132. Y. Yang, D. Henderson, P. S. Crozier, R. L. Rowley, and D. D. Busath, *Mol. Phys.*, **100**(18), 3011 (2001). Permeation of Ions Through a Model Biological Channel: Effect of Periodic Boundary Conditions and Cell Size.

133. B. Nadler, T. Naeh, and A. Schuss, *SIAM J. Appl. Math.*, **63**(3), 850 (2003). Connecting a Discrete Ionic Simulation to a Continuum.

134. B. Corry, M. Hoyles, T. Allen, M. Walker, S. Kuyucak, and S.-H. Chung, *Biophys. J.*, **82**(4), 1975 (2002). Reservoir Boundaries in Brownian Dynamics Simulations of Ion Channels.

135. C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation*, Institute of Physics Publishing, Philadelphia, Pennsylvania, 1991.

136. R. S. Eisenberg, *J. Membr. Biol.*, **150**(1), 1 (1996). Computing the Field in Proteins and Channels.

137. A. Syganow and E. von Kitzing, *Biophys. J.*, **76**(2), 768 (1999). (In)validity of the Constant Field and Constant Currents Assumptions in Theories of Ion Transport.

138. B. Roux and S. Bernèche, *Biophys. J.*, **82**(3), 1681 (2002). On the Potential Functions used in Molecular Dynamics Simulations of Ion Channels.

139. D. L. Ermak, *J. Chem. Phys.*, **62**(10), 4189 (1975). A Computer Simulation of Charged Particles in Solution. I. Technique and Equilibrium Properties.

140. P. Turq, F. Lantelme, and H. L. Friedman, *J. Chem. Phys.*, **66**(7), 3039 (1977). Brownian Dynamics: Its Application to Ionic Solutions.

141. W. F. van Gunsteren and H. J. C. Berendsen, *Mol. Phys.*, **45**(3), 637 (1982). Algorithms for Brownian Dynamics.

142. L. Verlet, *Phys. Rev.*, **159**(1), 159 (1967). Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules.

143. E. Jakobsson, *Method. Enzymol.*, **14**, 342 (1998). Using Theory and Simulation to Understand Permeation and Selectivity in Ion Channels.

144. P. S. Crozier, D. Henderson, R. L. Rowley, and D. D. Busath, *Biophys. J.*, **81**(6), 3077 (2001). Model Channel Ion Currents in NaCl-Extended Simple Point Charge Water Solution with Applied-Field Molecular Dynamics.

145. P. S. Crozier, R. L. Rowley, N. B. Holladay, D. Henderson, and D. D. Busath, *Phys. Rev. Lett.*, **86**(11), 2467 (2001). Molecular Dynamics Simulation of Continuous Current Flow Through a Model Biological Membrane Channel.

146. W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, *J. Chem. Phys.*, **79**(2), 926 (1983). Comparison of Simple Potential Functions for Simulating Liquid Water.

147. A. Wallqvist and R. D. Mountain, in *Reviews in Computational Chemistry*, Vol. 13, K. B. Lipkowitz and D. B. Boyd, Eds., VHC Publishers, New York, 1999, pp. 183–247. Molecular Models of Water: Derivation and Description.

148. H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, and J. Hermans in *Intermolecular Forces*, B. Pullman, Ed., Reidel, Dodrecht, the Netherlands, 1981, pp. 331–342. Interaction Models for Water in Relation to Protein Hydration.

149. W. L. Jorgensen, *J. Chem. Phys.*, **77**(7), 4156 (1982). Revised TIPS for Simulation of Liquid Water and Aqueous Solutions.

150. H. J. C. Berendsen, J. R. Grigera, and T. P. Straatsma, *J. Phys. Chem.*, **91**, 6269 (1987). The Missing Term in Effective Pair Potentials.

151. T. Schlick, in *Lecture Notes in Computational Science and Engineering*, Vol. 4, P. Deuflhard, J. Hermans, B. Leimkuhler, A. E. Mark, S. Reich, and R. D. Skeel, Eds., Springer, New York, pp. 227–262. Computational Molecular Dynamics: Challenges, Methods, Ideas.

152. M. Tuckerman, B. J. Berne, and G. J. Martyna, *J. Chem. Phys.*, **97**(3), 1990 (1992). Reversible Multiple Time Scale Molecular Dynamics.

153. X. Qian and T. Schlick, *J. Chem. Phys.*, **116**(14), 5971 (2002). Efficient Multiple-Time-Step Integrators with Distance-Based Force Splitting for Particle-Mesh-Ewald Molecular Dynamics Simulations.

154. T. Schlick, in *Interdisciplinary Applied Mathematics*, Vol. 21, Springer, New York, 2002. Molecular Modeling and Simulation.

155. S. Bernèche and B. Roux, *Biophys. J.*, **78**(6), 2900 (2000). Molecular Dynamics of the KcsA K^+ Channel in a Bilayer Membrane.

156. S. Bernèche and B. Roux, *Nature*, **414**, 73 (2001). Energetics of Ion Conduction Through the K^+ Channel.

157. T. P. Straatsma, in *Reviews in Computational Chemistry*, Vol. 9, K. B. Lipkowitz and D. B. Boyd, Eds., VHC Publishers, New York, 1996, pp. 81–127. Free Energy by Molecular Simulation.

158. S. Bernèche and B. Roux, *Proc. Natl. Acad. Sci. USA*, **100**(15), 8644 (2003). A Microscopic View of Ion Conduction Through the K^+ Channel.

159. S. W. Chiu, J. A. Novotny, and E. Jakobsson, *Biophys. J.*, **64**(1), 98 (1993). The Nature of Ion and Water Barrier Crossings in a Simulated Ion Channel.

160. A. B. Mamonov, R. D. Coalson, A. Nitzan, and M. G. Kurnikova, *Biophys. J.*, **84**(6), 3646 (2003). The Role of the Dielectric Barrier in Narrow Biological Channels: A Novel Composite Approach to Modeling Single-Channel Currents.

161. B. Roux, *Biophys. J.*, **77**(1), 139 (1999). Statistical Mechanical Equilibrium Theory of Selective Ion Channels.

162. G. M. Torrie and J. P. Valleau, *J. Comput. Phys.*, **23**, 187 (1977). Nonphysical Sampling Distributions in Monte Carlo Free-energy Estimation: Umbrella Sampling.

163. O. M. Becker, A. D. MacKerell, Jr., B. Roux, and M. Watanabe, Eds., *Computational Biochemistry and Biophysics*, Marcel Dekker, New York, 2001.

164. D. Beglov and B. Roux, *J. Chem. Phys.*, **100**(12), 9050 (1994). Finite Representations of an Infinite Bulk System: Solvent Boundary Potential for Computer Simulations.

165. M. Jalaie and K. B. Lipkowitz, in *Reviews in Computational Chemistry*, Vol. 14, K. B. Lipkowitz and D. B. Boyd, Eds., Wiley-VCH Publishers, New York, 2000, pp. 441–486. Published Force Field Parameters for Molecular Mechanics, Molecular Dynamics, and Monte Carlo Simulations.

166. T. A. Halgren and W. Damm, *Curr. Opin. Struct. Biol.*, **11**(2), 236 (2001). Polarizable Force Fields.

167. S. W. Rick and S. J. Stuart, in *Reviews in Computational Chemistry*, Vol 18, K. B. Lipkowitz and D. B. Boyd, Eds., Wiley-VHC Publishers, New York, 2002, pp. 89–146. Potentials and Algorithms for Incorporating Polarizability in Computer Simulations.

168. T. Allen, S. Kuyucak, and S.-H. Chung, *J. Chem. Phys.*, **111**(17), 7985 (1999). The Effect of Hydrophobic and Hydrophilic Channel Walls on the Structure and Diffusion of Water and Ions.

169. C. Domene and M. S. P. Sansom, *Biophys. J.*, **85**(5), 2787 (2003). Potassium Channel, Ions, and Water: Simulation Studies Based on the High Resolution X-Ray Structure of KcsA.

170. S. W. Rick, S. J. Stuart, and B. J. Berne, *J. Chem. Phys.*, **101**(7), 6141 (1994). Dynamical Fluctuating Charge Force Fields: Application to Liquid Water.

171. J. L. Banks, G. A. Kaminsky, R. Zhou, D. T. Mainz, B. J. Berne, and R. A. Friesner, *J. Chem. Phys.*, **110**(2), 741 (1999). Parametrizing a Polarizable Force Field from *Ab Initio* Data. I. The Fluctuating Point Charge Model.

172. H. A. Stern, G. A. Kaminski, J. L. Banks, R. Zhou, B. J. Berne, and R. A. Friesner, *J. Phys. Chem. B*, **103**(22), 4730 (1999). Fluctuating Charge, Polarizable Dipole, and Combined Models: Parametrization from *Ab Initio* Quantum Chemistry.

173. B. W. Arbuckle and P. Clancy, *J. Chem. Phys.*, **116**(12), 5090 (2002). Effects of the Ewald Sum on the Free Energy of the Extended Simple Point Charge Model for Water.

174. J. Hermans, H. J. C. Berendsen, W. F. van Gunsteren, and J. P. M. Postma, *Biopolymers*, **23**, 1513 (1984). A Consistent Empirical Potential for Water-Protein Interactions.

175. H. E. Alper and R. M. Levy, *J. Chem. Phys.*, **91**(2), 1242 (1989). Computer Simulation of the Dielectric Properties of Water: Studies of the Simple Point Charge and Transferrable Intermolecular Potential Models.

176. S.-H. Chiu, E. Jakobsson, S. Subramaniam, and J. A. McCammon, *Biophys. J.*, **60**(1), 273 (1991). Time-Correlation Analysis of Simulated Water Motion in Flexible and Rigid Gramicidin Channels.

177. A. Fick, *Annalen der Physik und Chemie*, **94**, 59 (1855). Über Diffusion.

178. C. W. Gardiner, in *Springer Series in Synergetics*, Vol. 13, Springer, New York, 1983. Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences.

179. P. E. Kloeden and E. Platen, in *Applications of Mathematics*, Corrected Third Edition, Springer-Verlag, 1999. Numerical Solution of Stochastic Differential Equations.

180. R. S. Eisenberg, M. M. Klosek, and Z. Schuss, *J. Chem. Phys.*, **102**(4), 1767 (1995). Diffusion as a Chemical Reaction: Stochastic Trajectories Between Fixed Concentrations.

181. D. A. McQuarrie, *Statistical Mechanics*, University Science Books, Sausalito, California, 2000.

182. S. A. Rice and P. Gray, in *Monographs in Statistical Physics and Thermodynamics*, Vol. 8, Wiley Interscience, New York, 1965. The Statistical Mechanics of Simple Liquids.

183. N. W. Ashcroft and N. D. Mermin, *Solid State Physics*, Holt–Sauders International Editions, Tokyo, 1981.

184. G. Baccarani and M. Wordeman, *Solid-State Electron.*, **28**, 407 (1985). An Investigation of Steady State Velocity Overshoot Effects in Si and GaAs Devices.

185. D. P. Chen, R. S. Eisenberg, T. W. Jerome, and C. Shu, *Biophys. J.*, **69**, 2304 (1995). Hydrodynamic Model of Temperature Change in Open Ionic Channels.

186. C. M. Snowden, *Introduction to Semiconductor Device Modelling*, World Scientific Publishing, Singapore, 1986.

187. H. K. Gummel, *IEEE Trans. Elect. Dev.*, **ED-11**, 455 (1964). A Self-Consistent Iterative Scheme for One-Dimensional Steady State Transistor Calculations.

188. D. L. Sharfetter and H. K. Gummel, *IEEE Trans. Elect. Dev.*, **ED-16**(1), 64 (1969). Large-Signal Analysis of a Silicon Read Diode Oscillator.

189. J. Molenaar, *Multigrid Methods for Semiconductor Device Simulation Technical Report CWI TRACT 100*, Center for Mathematics and Computer Science, Amsterdam, the Netherlands, 1993.

190. D. G. Levitt, *Biophys. J.*, **52**(3), 455 (1987). Exact Continuum Solution for a Channel that Can be Occupied by Two Ions.

191. A. E. Cardenas, R. D. Coalson, and M. G. Kurnikova, *Biophys. J.*, **79**(1), 80 (2000). Three-Dimensional Poisson-Nernst-Planck Theory Studies: Influence of Membrane Electrostatics on Gramicidin A Channel Conductance.

192. T. van der Straaten, S. Varma, S. W. Chiu, J. Tang, N. Aluru, R. S. Eisenberg, U. Ravaioli, and E. Jakobsson, in M. Laudon and B. Romanowicz, Eds., *Proceedings of the Second International Conference on Computational Nanoscience and Nanotechnology – ICCN2002*, San Juan, Puerto Rico (2002) pp. 60–63.

193. R. S. Eisenberg, *J. Membr. Biol.*, **150**, 1 (1996). Computing the Field in Protein and Channels.

194. B. Honig and A. Nichols, *Science*, **268**, 1144 (1995). Classical Electrostatics in Biology and Chemistry.

195. G. Lamm, in *Reviews in Computational Chemistry*, Vol. 19, K. B. Lipkowitz, R. Larter, and T. R. Cundari, Eds., wiley-VHC Publishers, New York, 2003, pp. 147–365. The Poisson-Boltzmann Equation.

196. T. Weiss, *Cellular Biophysics*, Vol. 1-2, MIT Press, Cambridge, Massachusetts, 1996.

197. B. Corry, S. Kuyucak, and S.-H. Chung, *J. Gen. Physiol.*, **114**, 597 (1999). Test of Poisson-Nernst-Plank Theory in Ion Channels.

198. B. Corry, S. Kuyucak, and S.-H. Chung, *Biophys. J.*, **78**(5), 2364 (2000). Tests of Continuum Theories as Models of Ion Channels. II. Poisson-Nernst-Planck Theory versus Brownian Dynamics.

199. G. Moy, B. Corry, S. Kuyucak, and S.-H. Chung, *Biophys. J.*, **78**(5), 2349 (2000). Tests of Continuum Theories as Models of Ion Channels. I. Poisson-Boltzmann Theory versus Brownian Dynamics.

200. B. Nadler, U. Hollerbach, and R. S. Eisenberg, *Phys. Rev. E*, **68**, 021905 (2003). Dielectric Boundary Force and Its Crucial Role in Gramicidin.

201. B. Corry, S. Kuyucak, and S.-H. Chung, *Biophys. J.*, **84**(6), 3594 (2003). Dielectric Self-Energy in Poisson-Boltzmann and Poisson-Nernst-Planck Models of Ion Channels.

202. R. S. Eisenberg, *Proceedings of the Biophysical Society Meeting*, Washington, DC (1993). From Structure to Permeation in Open Ionic Channels.

203. R. S. Eisenberg, in *Advanced Series in Physical Chemistry*, Vol. 7, New Developments and Theoretical Studies of Proteins. World Scientific, London, 1996, pp. 269–358. Atomic Biology, Electrostatics, and Ionic Channels.

204. R. J. Mashl, Y. Tang, J. Schnitzer, and E. Jakobsson, *Biophys. J.*, **81**(5), 2473 (2001). Hierarchical Approach to Predicting Permeation in Ion Channels.

205. M. F. Schumaker, R. Pomes, and B. Roux, *Biophys. J.*, **79**(6), 2840 (2000). A Combined Molecular Dynamics and Diffusion Model of Single Proton Conduction through Gramicidin.

CHAPTER 5

Wavelets in Chemistry and Cheminformatics

C. Matthew Sundling,[*] Nagamani Sukumar,[*] Hongmei Zhang,[*] Mark J. Embrechts,[†] and Curt M. Breneman[*†]

[*]*Department of Chemistry and Chemical Biology, Center for Biotechnology and Interdisciplinary Studies, Rensselaer Polytechnic Institute, Troy, New York*
[†]*Department of Decision Science and Engineering Systems, Rensselaer Polytechnic Institute, Troy, New York*

PREFACE

Wavelet transform methods developed quickly during the 1990s and have since become widely used in various fields of science and engineering, including many important applications in chemistry. The ability of wavelet methods to rapidly dissect signals into meaningful components makes them invaluable tools for data analysis and information compression. Unlike traditional signal processing methods, the wavelet transform offers simultaneous localization of information in both frequency and time or property domains. It makes the wavelet transform powerful in its ability to succinctly distill the details of complex and irregular property distributions or waveforms into meaningful and simple components. Consequently, wavelet transform methods are well suited to processing experimental data collected throughout the various areas of chemistry as well as to leading to new types of computationally generated molecular property descriptors. The diverse utility of wavelets has caused an explosion of application papers covering many areas of chemical

Reviews in Computational Chemistry, Volume 22
edited by Kenny B. Lipkowitz, Thomas R. Cundari, and Valerie J. Gillet

analysis, most of which land squarely in the realm of chemometrics or chemical spectroscopy, but significant applications in quantum chemistry, and recently, cheminformatics and computational chemistry, show how wavelets can be applied in any situation where data analysis is needed. Wavelet transform methods are a proven technology in signal cleaning and signal feature isolation and have provided chemists with better methods for analyzing and understanding their experimental data—by distilling *chemical information* from raw experimental or computational data. This chapter is a pedagogically driven overview of basic wavelet transformation methods and their applications in chemistry and cheminformatics.

INTRODUCTION TO WAVELETS

We provide here a brief introduction to the concept of wavelets, intentionally glossing over much technical and mathematical detail in favor of conveying a simple and conceptual perspective of wavelet techniques. For a thorough introduction to the theory and history of the wavelet transform, readers should examine the pertinent literature.[1–8] Other discussions of wavelets and their applications in chemistry may be found in Refs. 9–17. The predecessor of the wavelet transform (WT) is the Fourier transform (FT), which has a successful history of analyzing chemical spectra. In this section, we discuss the fundamentals of the Fourier transform to illustrate the advantages and disadvantages of this technique relative to wavelets and to show how the wavelet transform picks up where the Fourier transform leaves off. We also show that the wavelet transform acts as a ''mathematical microscope'' to reveal and focus on spectral features that are buried in the original signal but that are ignored by the Fourier transform. Wavelets thus allow for a thorough examination of the character of a signal, including high-frequency noise, asymmetric broad regions, short-term spikes, and other features of interest.

Throughout this chapter we will describe how wavelets can be used to analyze, clean, and encode molecular information in a dense and usable format, and we will show how wavelets provide a useful and stable means for representing molecular electronic property distributions for use in quantitative structure-activity relationship/quantitative structure-property relationship (QSAR/QSPR) modeling. Before that, however, some history of their place in signal analysis is appropriate.

Like the FT, the WT converts a signal from its normal time- or property-domain representation into another representation—in wavelet space—which reveals the frequency content of the original signal. A wavelet space representation not only separates frequency components, but unlike the FT, also gives their exact position and identifies their effective domain. It allows WT methods to separate, isolate, and analyze the individual components of a signal. The FTs and WTs are useful when analyzing many different types of chemistry

data, regardless of the domain to which they belong. Even though many important chemical "signals" exist within a variety of domains, we will discuss these methods in terms of time-domain signals, as well as their resulting frequency-domain and wavelet-domain transforms.

Fourier Transform

A time-domain signal contains a relationship between temporal information and amplitude information, but it gives no explicit frequency information. The key assumption in Fourier analysis is that a signal can be considered a composite of sinusoidal components, each having a specific frequency. The individual components are convoluted together in the original signal and are therefore generally immune to interpretation by direct inspection. FT converts the signal from a time domain into a frequency domain, giving us access to frequency and amplitude information, as illustrated in Figure 1. FT is a reversible process and gives two entirely different perspectives of the same data. However, we cannot get both the time-domain and frequency-domain information simultaneously, which is often critical for a variety of data analyses.

Continuous Fourier Transform

The continuous Fourier transform (CFT) of a real or complex continuous function is defined as

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t}dt \qquad [1]$$

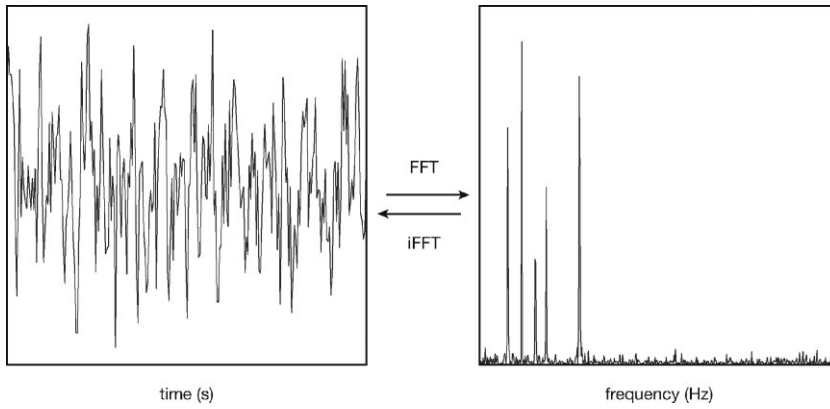


Figure 1 This illustration shows a signal converted from the time domain to the frequency domain using a Fourier transform technique. The fast Fourier transformation (FFT) is a discrete and computationally efficient version of the general Fourier transformation.

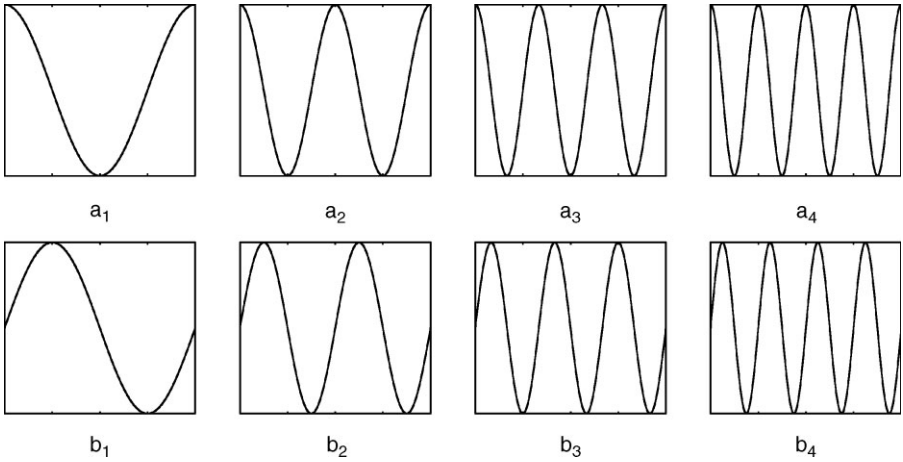


Figure 2 Illustrated are four example pairs of sinusoidal Fourier basis functions that constitute a portion of a Fourier series. Each pair of functions *a* and *b* has the same frequency but is 90° out of phase. They combine to give a series of terms of infinite domain, at particular frequencies, but of arbitrary phases.

and the inverse transform is defined as

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{+i\omega t} d\omega \qquad [2]$$

where

$$e^{-i\omega t} = \cos \omega t + i \sin \omega t \qquad [3]$$

In Eqs. [1–3], function $f(t)$ is a continuous time-domain function that is transformed into the frequency-domain function $\hat{f}(\omega)$.[1,4] Because our basis functions are sinusoidal, as defined by Eq. [3] and illustrated in Figure 2, a single component with frequency ω will affect the entire domain of the signal $f(t)$ equally. It makes the FT not entirely suitable for analyzing nonstationary (nonperiodic) functions (as depicted in Figure 3). Isolation of a given frequency component to a finite time region is necessary when dealing with localized signal features (such as a sharp spike) that are, by nature, nonstationary. The Fourier basis function entirely lacks the ability to distinguish or isolate time-domain information.

Short-Time Fourier Transformation

Given the limitations of the FT, some approximations are needed to handle nonstationary signals. The discrete FT (DFT) and the short-time FT (STFT, a.k.a. the Gabor transform) are two alternative transformation methods that address this issue.[1,3–5,9,18] In the mid-20th century, Jean Ville pointed out that

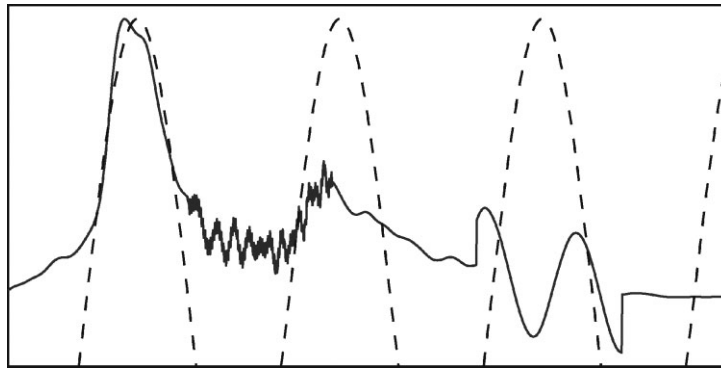


Figure 3 A nonstationary signal (solid line) is fitted with an FT basis function (dashed line). The basis function is fitted well against a single signal peak, but outside of the immediate region of the peak, the signal approximation suffers. It illustrates the ability of a Fourier basis function to isolate frequency information but not time-domain information.

two basic approaches to time-frequency analysis exist, but both attempt the same thing: to create a pseudo-stationary signal from a nonstationary one.[1] One approach filters different frequency bands and then splits these bands into pieces and analyzes their energy content. The other approach splits the signal into equal length sections in the time domain and then examines these pieces individually for their frequency content. The DFT uses angular sampling to isolate frequency bands, whereas STFT uses windowed-time frequency analysis.

In STFT, a nonstationary signal is divided into small windows in an attempt to achieve a locally stationary signal, as depicted in Figure 4. Each

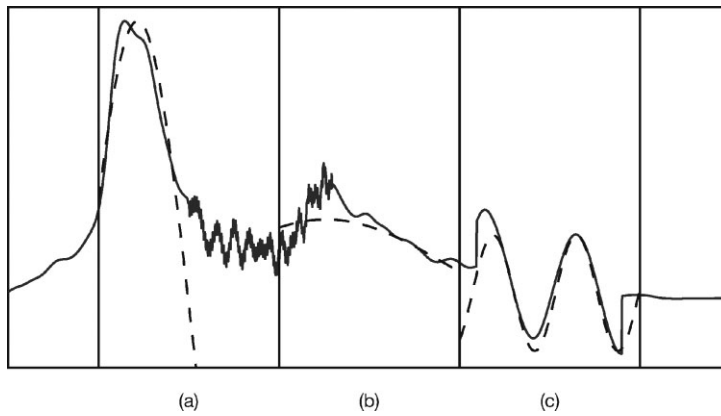


Figure 4 (*a*)–(*c*) The windowing of a nonstationary signal (solid line) in STFT analysis gives some locality of time information to the FT (dashed line is Fourier basis function). Even so, this method still suffers from a tradeoff of knowledge between time-domain and frequency-domain information.

window is analyzed with a normal FT to obtain the local frequency composition of a signal. The window length selection is crucial because it ensures that the local signal is sufficiently stationary. Selection of the STFT window width involves a tradeoff; however, narrowing the time window improves time/location resolution but reduces clarity of the frequency information. Naturally, the STFT becomes an FT if the window length is taken as infinity. Although the STFT procedure manages to garner some signal position information via the location of the windows, it is still fundamentally hindered by the stationary nature of the sinusoidal basis of the FT method. A transformation method that uses basis functions that are simultaneously localized in both the time and frequency domains would neatly avoid this tradeoff. The WT procedure does exactly this.

Wavelet Transform

The WT provides an entirely new perspective on traditional signal processing techniques (i.e., FT methods) for breaking up a signal into its component parts. Literally, the term ''wavelet'' means little wave. More specifically, a wavelet is a function that satisfies the following two conditions: (1) It has a small concentrated finite burst of energy in the time domain, and (2) it exhibits oscillation in time.[2] The Daubechies 6 wavelet, illustrated in Figure 5, clearly exhibits these characteristics. The first condition makes the wavelet ''little'' in the sense that it is well localized in the time domain. The second condition makes it periodic, giving it some wave-like character.

Given a particular wavelet function (a particular basis function), the wavelet transform operates in a manner similar to the FT by using its basis function to convert a signal from one domain to another. The WT

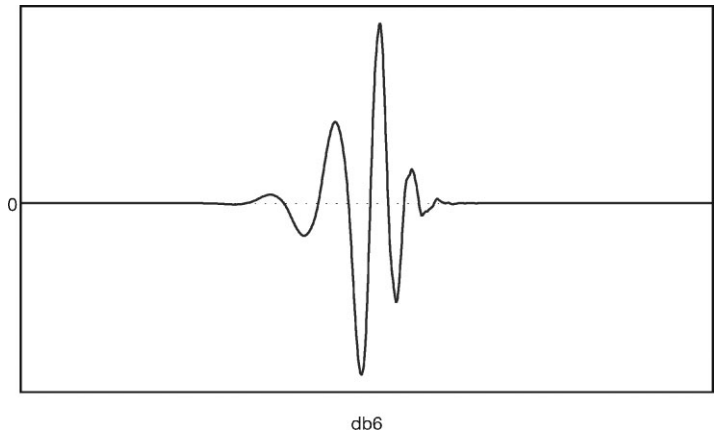


Figure 5 An example of a Daubechies wavelet (Daubechies 6) that illustrates the two interesting wavelet properties: localized in time and oscillation.

deconstructs a signal by using dilated (scaled) and translated (shifted) versions of the basis function. The attraction of WT techniques over FT techniques is that they localize information effectively in both the time and the frequency domains simultaneously.[6,9] Because of this, the WT is an ideal tool for analyzing nonstationary signals, such as spectra or molecular property distributions.

WT methods can be categorized into two main classes: (1) continuous WT (CWTs) and (2) discrete wavelet transforms (DWTs). Each is discussed below.

Continuous Wavelet Transform

The CWT is defined as

$$CWT[f(x)] = \int_{-\infty}^{+\infty} f(x)\Psi^*_{a,b}(x)dx \qquad [4]$$

and the inverse transform is defined as

$$f(x) = \frac{1}{C}\int_{0}^{+\infty}\int_{-\infty}^{+\infty} CWT[f(x)]\Psi_{a,b}(x)\frac{dadb}{a^2} \qquad [5]$$

where

$$\Psi_{a,b}(x) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{x-b}{a}\right), a, b \in, \mathbb{R}, a \neq 0 \qquad [6]$$

$$C = \int_{0}^{+\infty}\frac{\hat{\psi}^*(\omega)\hat{\psi}(\omega)}{\omega}d\omega \qquad [7]$$

$\Psi_{a,b}(x)$ is a dilated and translated version of the "mother wavelet" $\psi(x)$, where a is the scale, b is the translation, $1/\sqrt{|a|}$ is a normalization term, and * symbolizes complex conjugation. Equation [7] is the admissibility condition that gives C as a positive real number if the mother wavelet satisfies certain conditions (e.g., the integral of the mother wavelet equals zero).[1,3–5,9,11,18] There is no restriction on the choice of a and b in the CWT (other than $a \neq 0$), which means that the choice of $\Psi_{a,b}(x)$ is continuous in the time-scale domain. The wavelet can be positioned anywhere and scaled to any value for optimal fitting of the signal $f(x)$. The admissibility condition requires that only wavelets of certain character are capable of the reverse wavelet transform as described by Eq. [5]. It does not restrict the wavelets capable of the forward wavelet transform. In fact, many useful wavelet applications use wavelets that make the reverse transform impossible.

Although the FT separates the signal into a series of sine waves of different frequencies, the WT decomposes the signal into wavelets—dilated and

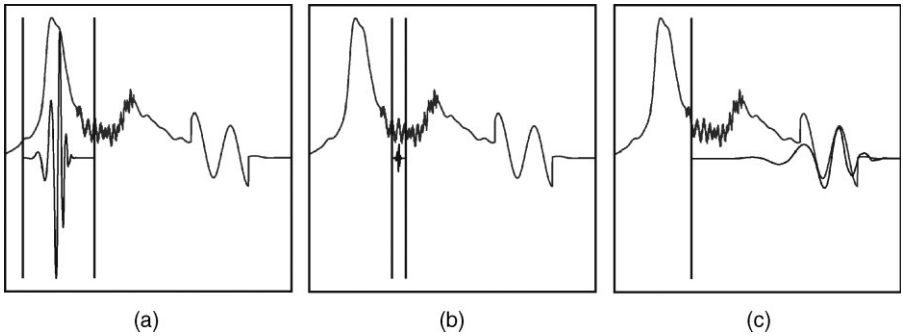


Figure 6 Shown here are (*a*)–(*c*) three wavelet ''windows'' over a nonstationary signal that illustrate a partial wavelet scaling analysis and how wavelets simultaneously identify component frequency and position information within the signal. The CWT performs an exhaustive fitting of the different features of the signal at different scales and positions of the wavelet function.

translated versions of the ''mother'' wavelet. Compared with a smooth and infinite sinusoidal wave function, the wavelet function is irregular in shape and compactly supported (i.e., has a limited domain where the function is nonzero). These properties make wavelets an ideal tool for analyzing nonstationary signals of finite length or duration; their irregular shape enables them to characterize signals with discontinuities or sharp changes, and their compact support enables them to represent signals with temporal or regional features. Figure 6 depicts how wavelets can fit and decompose a signal. By a variety of large and small dilations, the wavelets can be fit into the various kinks, shoulders, arcs, and spikes of a given signal. Scaling analysis allows us to process signals at different scales and resolution, elegantly revealing aspects of the signal that would be masked by the regularity of a sinusoidal wave. In fact, whereas the STFT attempts to force sinusoidal functions onto a time-localized section of the signal, wavelets scale naturally to represent a variety of different regions within a given signal.

The CWT is compactly described by Eqs. [4] and [6], but this definition allows for infinitely redundant transformations.[4,19] There is no limit to the number of dilated and translated wavelets ($\Psi_{a,b}(x)$, where a and b are real numbers) used in the transform. This unrestricted and unguided use of wavelets to convert a signal into wavelet space often prevents the use of an inverse wavelet transformation because of violations of the conditions required by Eq. [7]. Even though these transforms are redundant and nonreversible, they still reveal information about the character of a particular signal.

Techniques exist for reducing the redundancy in the CWT. These techniques isolate the dilated and translated wavelets to form the signal's ''skeleton.'' The skeleton wavelets consist of important features found in the original signal, called ridges, which are the defining curves of the waveform. By focusing

and restricting the wavelets to only these critical features, redundancy is removed. Such a small set of independent wavelets make these CWT adaptations practical, informative, and manageable.[6,19,20] Additionally, placing specific restraints on the selection of wavelet basis and on the dilation values (i.e., the a values) eliminates all redundancy in the transforms and gives rise to a fast and useful transformation method—the DWT.

Discrete Wavelet Transform

The main differences between the DWT and the CWT are specific requirements for the mother wavelets and the allowable dilation and translation values. Explicitly, Eq. [4] becomes

$$DWT[f(x)] = \int_{-\infty}^{+\infty} f(x)\Psi^*_{a,b}(x)dx\,, a = 2^i, \mathrm{i} \in \mathbb{N}, b \in \mathbb{N} \qquad [8]$$

where the dilation variable a is restricted to powers of two (i.e., $a = 2^i, i \in \{0, 1, 2, \ldots\}$), and our translation variable b is a whole number (i.e., $b \in \{0, 1, 2, \ldots\}$). These values are called dyadic dilations and translations. In principle, the CWT, with no restriction applied to the choice of these two coordinates, maps the entire "wavelet space" [i.e., the (a, b) plane]. The DWT confines us to specific nonredundant regions of the wavelet space.

Rather than having a continuum of wavelet dilations as in the CWT, the discrete wavelet transform uses discrete dilations that can be thought of as filters of different scales. These act as cutoff frequencies to divide the signal into different frequency bands.[21] Wavelets are actually a pair of filters, called the wavelet and the scaling function, as depicted in Figure 7. To separate each

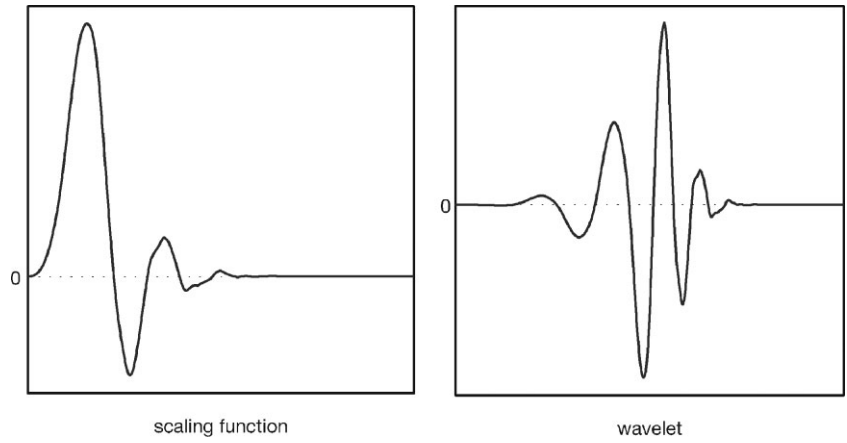


Figure 7 Each wavelet basis is actually a pair of functions: the wavelet and its scaling function. These are the two functions of the Daubechies 6 wavelet.

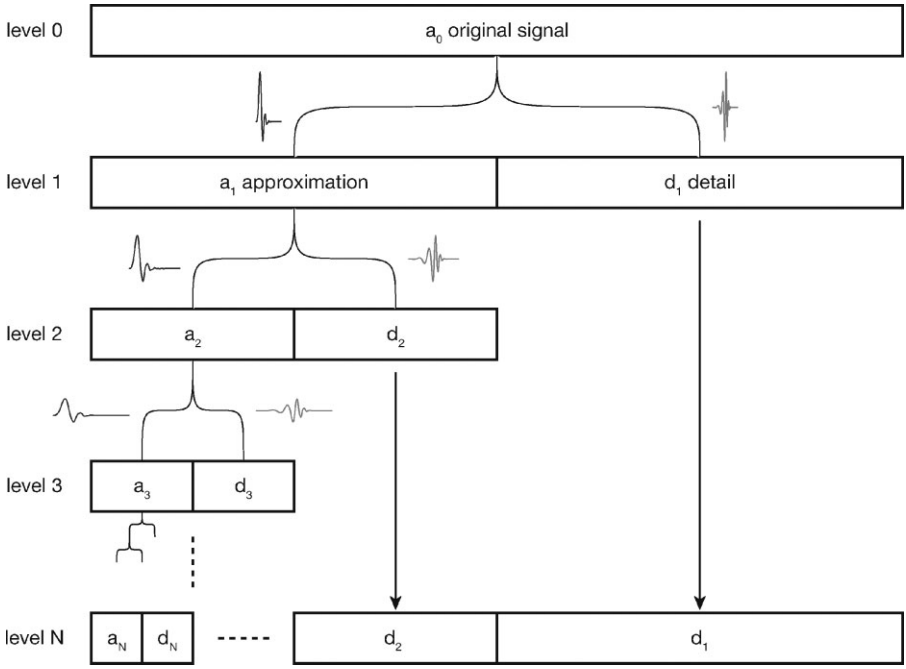


Figure 8 Overview of the DWT and multiresolution analysis scheme known as the pyramid algorithm.[7] The original signal is separated into low-frequency and high-frequency components, which comprise the signal approximation and detail information, respectively. Each level decomposes the approximation information further, making each level of detail (d_i) a separate frequency band.

frequency band, the wavelet is used to isolate the information pertaining to that band, and the scaling function separates out everything else. This process is part of a method called multiresolution analysis.[7]

Multiresolution analysis[3,4,7,19] is the most common DWT method, and it involves a hierarchy of low- and high-pass filters to successively separate the finer details from the remainder of a signal. The term "DWT" is often used to refer specifically to multiresolution analysis as implemented by the *pyramid algorithm*. Within the pyramid algorithm, the original signal is decomposed successively into components of lower frequency, whereas the high-frequency components are not analyzed further, as illustrated in Figure 8.[7] The analysis begins with two complementary filters, one low pass and one high pass, to separate the high-frequency details from the rest of the signal. Then, with each transformation step, increasingly coarse information is separated from the remaining portion of the signal (see Figure 9). The maximum number of dilations/separations that can be performed is determined by the input size of the data being analyzed.

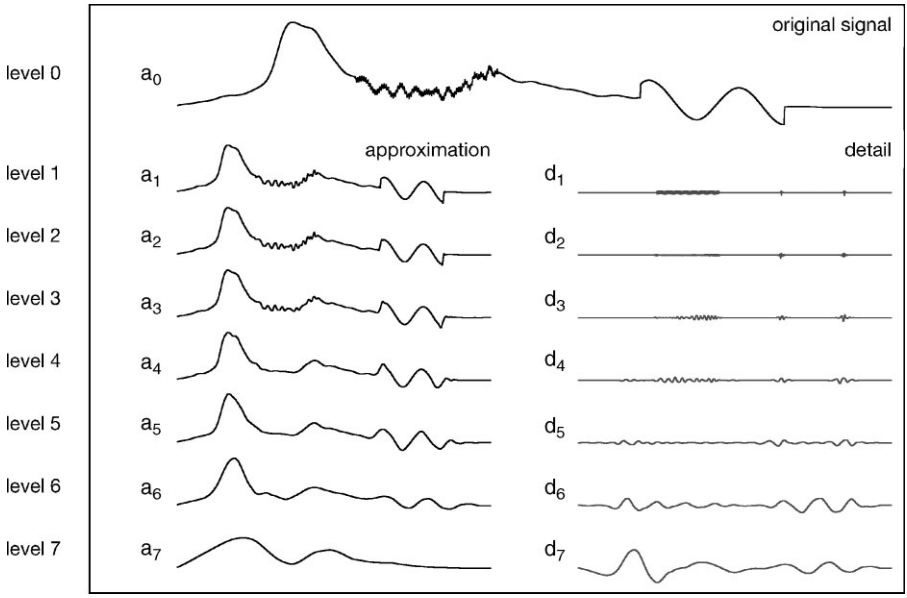


Figure 9 This series illustrates multiresolution analysis, separating out the high-frequency information at each level of transformation in the pyramid algorithm (illustrated in Figure 8). Note that the approximation (a_i) signal in higher level iterations contain much less detailed information, because this has been removed and encoded into wavelet detail coefficients at each DWT deconstruction step. To reconstruct the original signal, the inverse DWT needs the wavelet coefficients of a given approximation level *i* (a_i) and all detail information leading to that level ($d_{1\text{-}i}$).

Specifically, given a signal represented by a series of 2^N values, the signal is divided into N discrete levels of detail. The transformation turns the original signal into a set of 2^N wavelet coefficients, which reflect the individual contributions of their associated scaled wavelet. The original signal is really a combination of the dilated and translated wavelets as prescribed by the wavelet coefficients.

The restrictions placed on the mother wavelets for multiresolution analysis do not limit the variety of shapes that can be used as mother wavelets; different researchers have proposed several different wavelet functions, each with benefits and drawbacks.[3] The wavelet shape tradeoff is between how compactly it can be localized in space and its level of smoothness. For example, the Haar wavelet, which is the simplest wavelet and was identified almost 100 years ago,[22] is well localized in space, but it has an ''unnatural'' square-wave oscillation (see Figure 10). Many related wavelets exist, collectively referred to as wavelet families;[6] some of these families include the Meyer wavelet, Coiflet wavelet, spline wavelet, orthogonal wavelet, symmlet wavelet, and local cosine basis. Figure 10 depicts several of these wavelets and

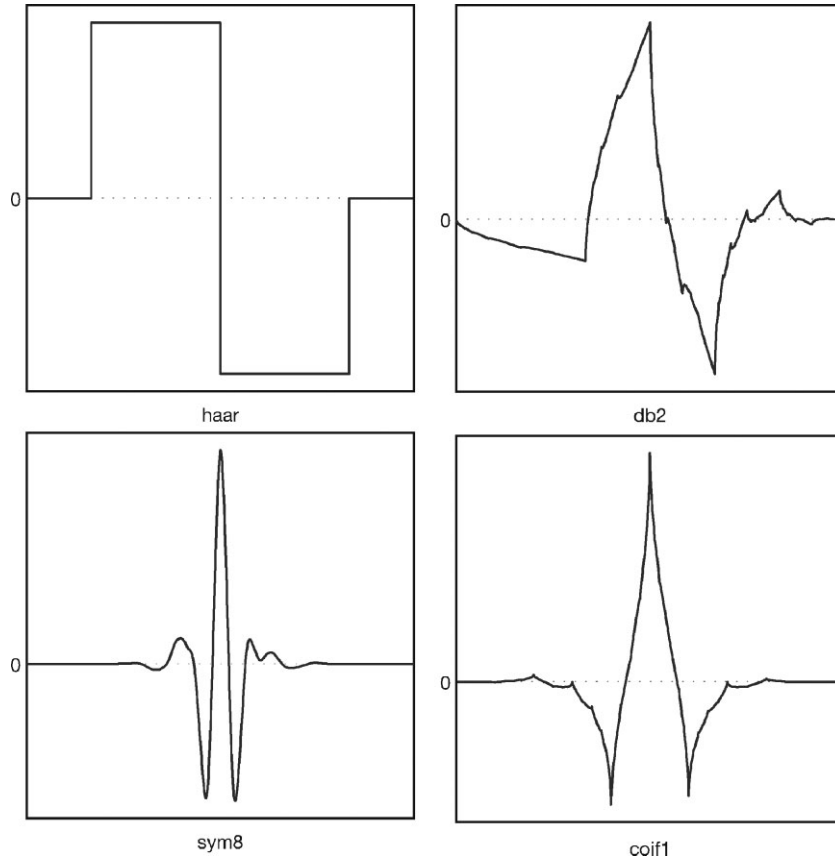


Figure 10 Some families of wavelets used for multiresolution analysis: (a) Haar, (b) Daubechies 2, (c) Symmlet 8, and (d) Coiflet.

illustrates some of their features. Depending on the shape of the original signal and how it is being analyzed, some wavelet functions will outperform others. Wavelet selection is, consequently, an important facet of wavelet analysis. For example, a wavelet basis with sharp, narrow peaks may be better suited to transform and characterize a particularly disjointed signal, whereas a simple, gentle wavelet may be more appropriate for a smoother signal. Selecting an optimal wavelet adapted to a particular type of signal allows for finer, more economical separation of signal features. In a general sense, an optimal wavelet basis function would concentrate the signal features to a small number of large-valued wavelet coefficients. This leads to interesting and useful applications of the wavelet transform, such as signal compression and feature isolation, both of which are important in various chemical applications and are examined further in the following sections.[15]

Wavelet Packet Transform

The wavelet packet transform (WPT) is a generalized version of the pyramid algorithm[8,23] in which the signal is successively separated into the low-pass information and "the rest" of the signal. In contrast to the pyramid algorithm used in DWT, both the low-pass and the high-pass information are iteratively transformed in WPT, creating a complete tree as compared with the single branch enumerated by the pyramid algorithm. Comparing Figures 11 and 8 illustrates this difference. As mentioned, optimal wavelet function selection is important for obtaining an efficient representation of the signal in wavelet space. The wavelet packet transform takes this notion of efficiency even further by offering the flexibility of choosing the final signal representation, which is also known as the "signal basis."[24] With the full hierarchy evaluated, we have a choice of combining different levels of high-pass and low-pass filtering, so we can select a signal representation that is most suited to our needs.

Selection of the "best" basis or representation from the WPT hierarchy means choosing a combination of orthogonal, nonredundant coefficients from

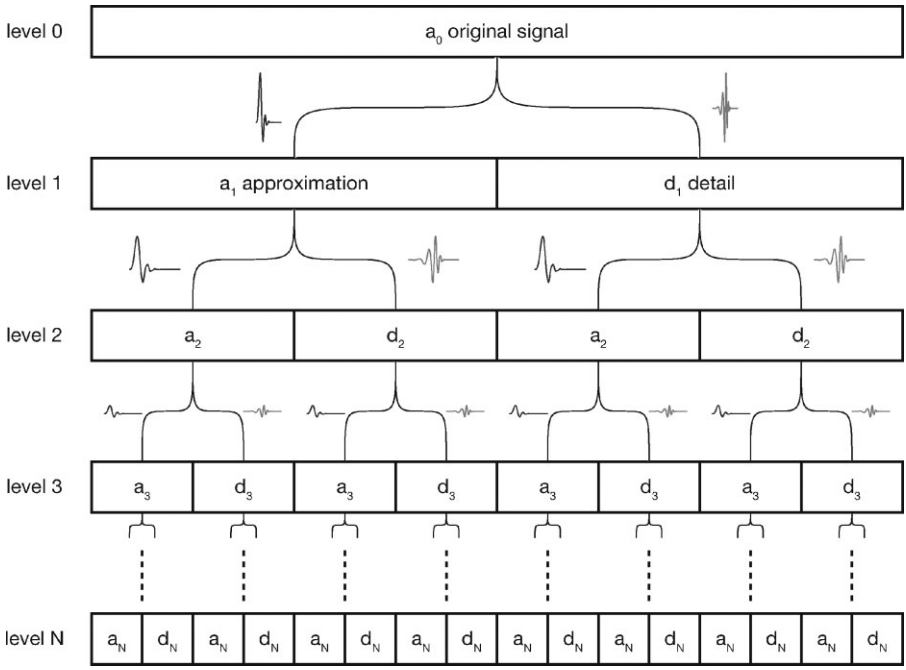


Figure 11 Illustrated here is the WPT. The pyramid algorithm enumerates a single branch (see Figure 8), whereas the WPT enumerates the complete tree of iterative decompositions. At each level, both the approximation and the detail information are separated into low-frequency and high-frequency components.

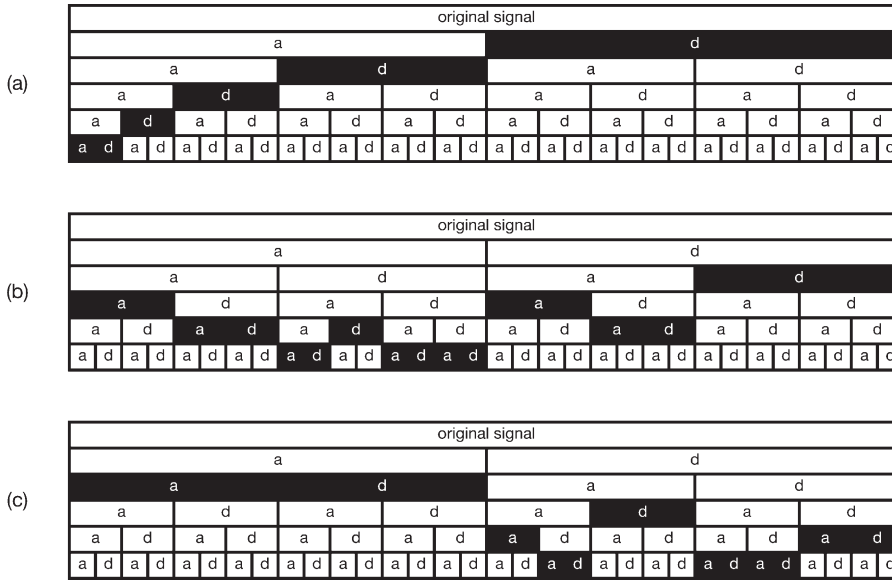


Figure 12 The signal representation or basis of the (a) pyramid algorithm and two examples (b, c) of selected representations from the entire hierarchy computed by the WPT are shown. The WPT allows for optimal signal basis selection by enumerating the complete decomposition tree, so that all signal representations can be evaluated.

the different scale levels of the WPT. An entropy-based algorithm developed by Coifman and Wickerhauser[24] provides a quantitative criterion for selecting a basis for signal representation at the lowest information cost. Although the pyramid algorithm blindly follows a single branch of the WPT tree and only computes a single combination of the many possible bases (see Figure 12), the WPT procedure affords the flexibility of choosing the optimal signal representation for any application.

Wavelets vs. Fourier Transforms: A Summary

The ability of WTs to resolve a signal into its component features makes it useable for many practical applications. The FT, while revealing the frequency characteristics of a signal, is limited by the assumption that all signals are stationary (periodic). It, in turn, provides no resolution in the time domain. The power of the wavelet transform is in its ability to succinctly distill the details of complex and irregular signals into meaningful and simple components—in essence, to encode both the time (or any domain) and the frequency information simultaneously.

APPLICATION OF WAVELETS IN CHEMISTRY

Smoothing and Denoising

Experimental data always contain noise from various sources, and it must be isolated, understood, and usually removed before effective data analysis can proceed. The goal of smoothing and denoising techniques is to separate useful from useless information. These techniques can be tailored to remove the noise from spectra, improve the signal-to-noise ratio in analytical images, improve resolution of peaks by removing background signals, and give clarity and focus to numerous types of data with different kinds of problems. Although popular cleaning techniques such as Savitzky–Golay smoothing and Fourier filtering[25–28] could be employed for these tasks, the WT-based signal cleaning techniques have grown in popularity because of their efficacy and utility in handling noise.

In general terms, smoothing and denoising are used to isolate and remove noise or background signals and to resolve features and peak shapes.[29] Smoothing and denoising are different, albeit related: smoothing is the process of removing the high-frequency components of a signal, regardless of their amplitude, and denoising is the process of removing low-amplitude components, regardless of their frequency.[25] All types of signal noise exist in experimental data, the most common of which can be found in the high-frequency range, especially when considering either time-based data or spectral data. Background signals, another type of noise, are often found in the low-frequency range. Although knowing the location (frequency and position) of noise in a signal is important, it is also important to understand the type or character of the noise. Heteroscedastic noise, or noise of changing variance, must be diagnosed and is usually handled differently from the more common homoscedastic noise. The treatment of heteroscedastic noise is a challenging problem for any signal smoothing technique, including WT methods. An example of heteroscedasticity is when the variance of noise increases as the overall signal strength increases.

Wavelet transform methods, particularly multiresolution analysis (MRA) methods, decompose a signal into frequency bands. That separation allows for the targeting of frequencies of particular interest or, in the case of noise, disinterest to the researcher. A signal may be considered to be a combination of component wavelets, so reducing the contribution of ''noise wavelets'' will result in a clean, smooth signal. The algorithm for smoothing a signal (see Figure 13) consists of four steps: (1) transform the signal, (2) isolate the wavelet coefficients corresponding to the high-frequency components, (3) ''zero-out'' or reduce these coefficients, and (4) apply a reverse wavelet transform to the signal. The final smooth signal will have the original features of interest without the high-frequency noise. Figure 13 illustrates the smoothing routine using the pyramid algorithm, but the routine is easily

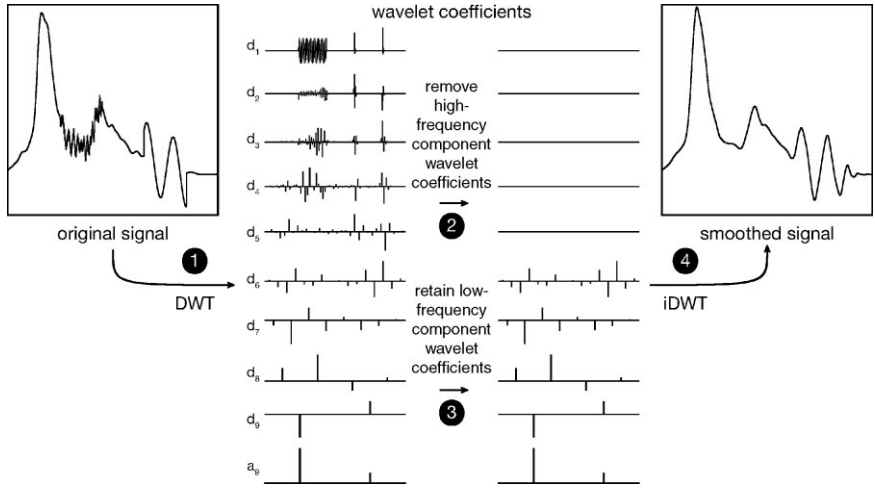


Figure 13 WT-based smoothing has four steps: (1) Transform the signal, (2) isolate the wavelet coefficients corresponding to the high-frequency components, (3) ''zero-out'' or reduce these coefficients, and (4) apply a reverse WT to the signal. Compare this with the denoising routine illustrated in Figure 14.

modified for use with the WPT, often yielding better results.[23] WPT not only separates the signal into frequency bands for noise isolation, but it also gives the optimal basis for signal representation. It means our signal is represented by the most sparse, compact basis of wavelet coefficients, focusing the signal features into a few large wavelet coefficients. It allows for confident and localized modification of frequency band information, with improved noise suppression and minimal cross-band effects when removing small wavelet coefficients.

The denoising algorithm is similar to the smoothing algorithm, except that in step 3, the small-amplitude coefficients are targeted for removal regardless of frequency (illustrated in Figure 14); this is sometimes referred to as wavelet thresholding.[29] WT denoising methods can retain the interesting and often subtle features normally destroyed by other aggressive smoothing techniques, including maintaining edge sharpness and peak shapes.[30] Donoho formalized the process of denoising and showed that if all wavelet coefficients are moved toward zero by proportional amounts, the resulting signal will then be similar to the original signal.[31] This approach virtually eliminates the small wavelet coefficients (and their encoded noise information), while retaining the important wavelets that contribute most to the shape of the original signal. Although variations exist of basic smoothing and denoising methods tailored to specific types of noise,[32] the core concept of wavelet smoothing and denoising techniques remains the same.

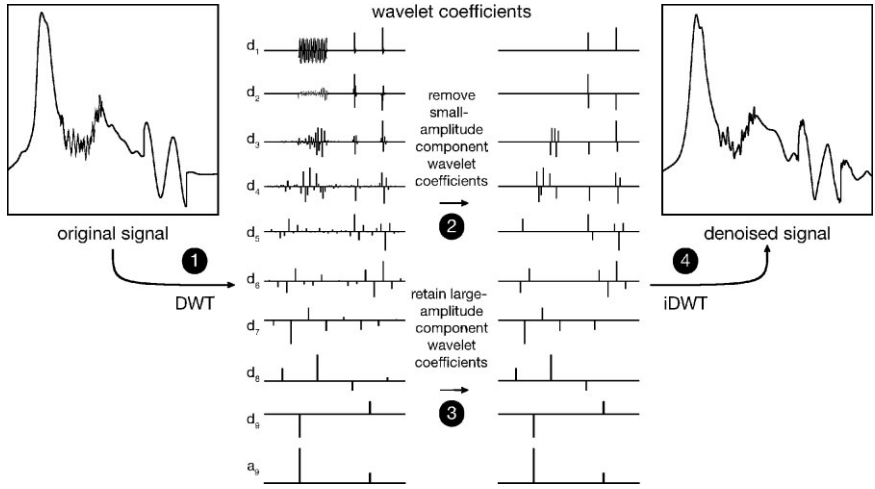


Figure 14 WT-based denoising has four steps: (1) Transform the signal, (2) isolate the small-amplitude wavelet coefficients corresponding to the noise components, (3) "zero-out" or reduce these coefficients, and (4) apply a reverse WT to the signal. Compare this with the smoothing routine illustrated in Figure 13. The isolation of small-amplitude coefficients in step 2 was achieved by using a progressive reduction hard-thresholding approach, which reduces the elimination threshold for each lower frequency band.

Variations of the wavelet smoothing and denoising routines described above have been thoroughly investigated and tested by Mittermayr et al. against more traditional methods, such as Savitzky–Golay smoothing and Fourier filtering.[27] Other investigators were rigorous in their evaluation of WT smoothing techniques; they used the best alternative smoothing techniques available and evaluated optimally selected mother wavelets tailored to different types of data.[25,26,28,33] Their analysis showed good results for wavelet denoising, outperforming the more traditional techniques on a variety of experimental data. There is an extensive list of publications within the chemistry literature, which are reviewed in Refs. 9–10, where WT cleaning of data has improved data analysis. In all cases, the goal of smoothing or denoising a signal was to simplify and clarify experimental data. Other examples of wavelet-based smoothing and denoising may be found in a multitude of applications including chromatography,[17,26,34,35] infrared spectroscopy (with and without heteroscedastic noise),[33,36–38] ultraviolet-visible spectroscopy,[39] mass spectrometry,[30,40–43] voltammetry,[39,44–46] capillary electrophoresis,[28] molecular superposition methods,[47] and photoacoustic spectroscopy.[48]

Heteroscedastic backgrounds can confound signals in a way that simple wavelet thresholding routines become ineffective at removing their influences. The changing variance of this noise allows the noise to move from one

frequency band to another. Simple wavelet thresholding of the DWT is not sufficient for handling noise that moves between frequency bands, but WPT methods allow for intelligent isolation of the background. WPT denoising techniques were used to remove heteroscedastic noise from near-infrared spectra[38] and from GC/MS spectra;[43] in both cases, the resolution of overlapping peaks having low signal-to-noise ratio was improved. Although the CWT can isolate the heteroscedastic noise, it is a cumbersome approach and is not often implemented in practice.[38]

Signal Feature Isolation

Feature isolation is a general form of signal analysis. It concerns the broader problem of understanding and quantifying a signal and its component features rather than removing one particular type of feature from a signal (e.g., noise). These features often include the location and size of sharp spikes, critical points, smooth regions, discontinuities, and frequency composition.[49–54]

Multiresolution analysis (MRA) techniques, which include the DWT and WPT methods, effectively divide a signal into frequency bands using wavelet functions of known position, which give the precise locations of all signal features in wavelet space. Knowing the locations of specific components of the signal allows them to be analyzed, enhanced, cleaned, or removed. MRA methods have the additional advantage of giving an orthogonal representation of the original signal, which allows for local modifications or "feature selection" without introducing the global changes that could otherwise affect the signal. Both the smoothing and the denoising routines are examples of this application.

Although MRA methods are useful in many applications, the CWT is often more accurate for signal characterization. The CWT does not enjoy the benefit of orthogonality, but it is not restricted in its placement of wavelet functions during the transform. The CWT can use optimally dilated and translated wavelets to represent every crevice and region of a signal to locate features of interest precisely. This allows the CWT to recognize, with great precision, where important signal shape features occur, such as peaks, kinks, smooth regions, and edges. For example, signal critical points can be located, dramatic changes in frequency content can be identified, and the behavior of the derivative of a curve can be characterized. The penalty for not having an orthogonal representation of the signal is that CWT-based methods cannot manipulate signals as conveniently as can MRA methods: Nonetheless the redundant signal representation inherent in CWT methods increases the precision and resolution of the signal analysis. In general, MRA methods are used to clean, change, or clarify a signal, whereas CWT methods are used to perform precise and thorough signal diagnostics.

WT-based methods for signal isolation and analysis have improved significantly the deconstruction, quantitative analysis, and cleaning of many

types of experimental data[55,56] ranging from chromatography,[57–64] infrared spectroscopy,[65–72] ultraviolet-visible spectroscopy,[67] mass spectrometry,[73] x-ray absorption,[74,75] nuclear magnetic resonance,[76–80] to other studies.[81,82] Other interesting applications use methods of wavelet-based image fusion[52,83–87] to combine data obtained from different sources for better analysis and enhanced information extraction.[88–90]

Signal Compression

The purpose of data compression is to reduce storage space and to concentrate signal information. The central concept of wavelet compression is to represent data in an optimal manner, so that only a small number of wavelet coefficients are needed to capture most of the original signal.

Many specific compression routines using the WT exist, but all are similar in that they move a signal from a higher dimensional space to a lower dimensional subspace. Compression takes advantage of the WT because it inherently focuses the main features and overall shape of a signal into a relatively small number of wavelet coefficients. Data compression using the WPT[8] starts by separating a signal into its frequency bands using multiresolution analysis. The best signal basis is then selected such that it minimizes the number of wavelet coefficients, which best represents the original signal. The best-basis selection can be achieved with a variety of optimizing functions. Normally an information entropy metric is used to guide the selection of a maximally representing but minimum-length basis.[91] Compression is achieved when fewer variables or data values are required to represent the original signal. The selection of the best basis is usually optimized for each signal, but there is an advantage to selecting the best basis for a set of signals simultaneously,[8,92] especially when creating a library of similar data, such as chemical spectra or molecular property information.

Data compression is a useful, if not novel, application of wavelets in chemistry. It has found great use in spectral compression and storage and stands as an important application for data archiving. WPT-based data compression is useful for archiving infrared spectral libraries,[93] specifically when a large amount of experimental data must be quickly and accurately processed and stored. Infrared spectra contain sharp peaks and features, making them well suited for compression by unsupervised DWT with basic wavelet-thresholding techniques. It is basically a denoising approach to compression, in that removing the noise coefficients saves storage space.[94]

A second approach to data compression is to compress infrared spectra with a construct called a wavelet neural network (WNN).[95] The WNN approach stores large amounts of infrared data for fast archiving of spectral data. It is achieved by modifying the machine learning technique of artificial neural networks (ANNs)[96] to capture the shape of infrared spectra using wavelet basis functions. The WNN approach is similar to another approach

used to store ultraviolet visible spectral information that is presented in Ref. [93]. Although the ANN approaches are very effective for storing spectral data, it is limited to operating on an expected type of data (i.e., infrared spectra), and it is not a general compression approach for any free form signal, image, or otherwise.

Quantum Chemistry

WT methods are flexible, robust, and particularly useful for representing complex and intricate functions, making them an interesting alternative to other, more common basis sets for representing molecular wave functions. Quantum chemical applications typically involve the computational evaluation of many-electron molecular wave functions (or electron density distributions), through approximate solutions of the Schrödinger equation. Semiempirical and ab initio quantum mechanical methods are commonly used to compute electronic structure and molecular properties derived from the wave function, as well as spectroscopic[97] properties originating from transitions between energy levels.[98–100]

The electronic wave functions of molecules vary much more dramatically near atomic nuclei than in the interatomic spaces.[101] It makes the computational cost of maintaining high-resolution wave functions near the nuclei much higher than when representing valence regions further away from any nuclei. In traditional ab initio approaches for representing electronic structure, a molecular wave function is expanded in a series of basis functions, typically as a linear combination of Gaussian functions. Because the expansions are uniform, adjusting the approximation to give improved resolution near atom centers requires a dramatic increase in the number of terms through the introduction of double- and triple-zeta basis sets.[102] An alternative approach for increasing local resolution in highly nonlinear wave functions is to use specialized basis sets that depend explicitly on their locations relative to nuclear positions. The spatially localized property of wavelets can be useful for creating a consistent basis function for electronic distribution calculations.[103,104] Wavelets can characterize the highly nonlinear wave functions encountered in quantum chemistry because they can adjust to fit widely varying nonstationary functions.[105] Wavelets have also helped to create well-behaved and consistent descriptions of the properties of electron density distributions.[106]

Iyengar and Frisch[107] have demonstrated the fundamental equivalence between the wavelet theory of multiresolution analysis and the translation and dilation operations on the primitive Cartesian Gaussian basis functions used in electronic structure theory:

$$\chi^{R}_{l,m,n}(r) = (x - R_x)^l (y - R_y)^m (z - R_z)^n e^{-\alpha(r-R)^2}, l, m, n \in \mathbb{N} \qquad [9]$$

where

$$R = \{R_x, R_y, R_z\} \quad [10]$$

The positive integers l, m, and n determine the orbital angular momentum of the basis function, and R is the Gaussian center. Thus, the Gaussian basis function χ has a translation property represented by its dependence on the position of atom center R as well as a dilatory property, wherein for any given value of l, m, n, and R, the exponent α may have multiple values; the primitive Gaussians with smaller α being simply dilated versions of the original—see Eq. [6] and Figure 6. These authors have shown that primitive Gaussians are, in fact, multiwavelets with nonintegral scale factors. The Gaussian multiwavelet basis is nonorthogonal on account of these nonintegral scale factors, giving rise to different levels of basis set completeness at different molecular geometries. Hence, the quality of a Gaussian basis set changes as the nuclei move—a well-known artifact of ab initio quantum chemistry known as the basis set superposition error (BSSE).[108] This analysis provides new ways to ascertain and control the quality of a basis set during ab initio molecular dynamics.

Modisette et al.[109] have illustrated how wavelets can improve both resolution and accuracy over traditional ab initio methods. A three-dimension wavelet analysis was used for electronic structure calculations by Cho et al.[97] They took advantage of the stable nature of the WT to provide a systematically improvable and tractable description of electronic wave functions, thereby overcoming some limitations of conventional basis set expansions. It was demonstrated by computing the 1-s states for all naturally occurring nuclei in the periodic table from hydrogen to uranium, as well as their interaction energies with the hydrogen molecule ion. Another study investigated position and momentum information from solutions of the Hartree–Fock equation and found wavelet-based analysis provided more information concerning the oscillatory nature of a time-dependent wave function than did traditional FT approaches.[110] These authors found that wavelets were valuable for improving current methods for total energy calculations.[111] Wavelets have also been used to improve empirical force field representations for studying the behavior of biological macromolecules.[112]

Harrison, et al. have reported an efficient, accurate multiresolution solver for the Kohn–Sham[113] and Hartree–Fock[114,115] self-consistent field methods for general polyatomic molecules. The Hartree–Fock exchange is a nonlocal operator, whose evaluation has been a computational bottleneck for electronic structure calculations, scaling as $O(N^{3-4})$ for small molecules and no better than $O(N^2 \log N)$ for larger systems. Although earlier applications of wavelets and multiresolution analysis to quantum chemistry employed single-component smooth wavelets, these authors used sparse multiwavelet bases and localized molecular orbitals to attain near-linear scaling in electronic structure computations.

Classification, Regression, and QSAR/QSPR

Understanding the relationship between the structure of a molecule and its physicochemical properties is vital for compound development and property optimization efforts. It is especially true for the expensive task of screening molecular databases for specific biological activities and developing new therapeutic lead compounds. Modern methods of rational drug design depend heavily on the use of computer models to better understand the relationships between compound structures and pharmacokinetic and biochemical behavior. The technique of QSAR modeling developed from this need and seeks correlations between molecular structure and observable molecular properties. There is a logical disconnect in the way molecules need to be represented to be understandable to chemists, and the way they must be represented for machine learning applications. To appropriately represent molecules for numerical analysis, the important features of each compound must be summarized by a concise set of descriptors.[116–118] The existing body of QSAR literature attests to the effectiveness of this technique.[119] Although the term "QSAR" is normally associated with models developed to explain the properties of small drug-like molecules, it is often used to describe the broader field of QSPR modeling that is used in chemometrics, cheminformatics, and analytical chemistry.

Quantifying the relationship between a molecule and its properties is an important step toward understanding and predicting behavior, whether the models refer to experimental spectra, sensor responses, or a set of molecular descriptors. Classification and regression are two main types of modeling approaches frequently used in QSAR/QSPR analysis: *Classification* entails assigning data to a discrete category, or clustering it into similar classes, whereas *regression* forms a continuous model that estimates the magnitude of molecular responses. Both approaches use machine learning methods taken from the fields of statistics and computer science, and both seek to refine raw data into an understandable form. It usually means that a simplification of the raw data is required to reveal important discriminatory features within the data.

Data refinement is important because it solves two problems that are frequently encountered while building QSPR models. The first problem is the "curse of dimensionality," and the second is "data variance" within the raw data. These problems are observed while building two common types of QSPR models that appear in chemometrics: pattern recognition of spectroscopic and chromatographic data. The first problem—the "curse of dimensionality" often originates when the number of features or dimensions used as input to a model greatly exceeds the number of cases or data points available for model development. Raw, continuous spectral data consist of many variables,[120] the use of which can result in a dimensionality problem, particularly when using high-capacity modeling methods such as ANNs. It means that the quality and importance of each variable is reduced dramatically, which in turn can affect

modeling adversely.[121] The many variables or dimensions can inundate a complex mathematical model with features, allowing the model to locate spurious, even false relationships between a molecule and a molecular property.[122] The second problem with using raw data is "data variance," which means that the model can suffer from unstable conditions such as changing noise variance or shifting relative peak positions.[123] It also originates as the classic problem of *data alignment* observed in both two-dimensional and three-dimensional molecular QSAR modeling. Such variability between related signals can render the data incomprehensible to classification schemes and regression methods. In general, models tend to increase in complexity with irrelevant sources of variance and noise,[124] making them less general, less robust, and less accurate. An example where data variance harms pattern recognition is observed with infrared spectra: When taking spectra from two different spectrophotometers, the precise peak position or signal-to-noise ratio may be sufficiently different to confound pattern recognition routines—even when the infrared spectra are of good quality. Instead of using such raw data directly, preprocessing is often required to convert the data to a more useable form for effective model building.

WT methods offer effective ways to address the two main problems of capturing the information contained within raw data. First, they facilitate feature/dimension reduction by converting raw data into a more succinct representation in wavelet space. The wavelets isolate and concentrate the shape and character of the raw signal into a relatively small number of wavelet coefficients. Using the wavelet coefficients as the data features (descriptors) affords data representation with a dramatic reduction in the number of variables.[122,125,126] Second, the use of feature isolation and extraction methods removes variance and error from the raw data giving standardized and consistent data representations.[127] These methods often include basic signal smoothing and denoising, but more complicated data cleaning, such as removal of the variance in peak positions, is possible as well.[128] The removal of unnecessary information from the raw data further concentrates the desired chemical information in the remaining variables used to represent the signal. What remains is a highly compact, consistent, and standardized representation of the important discriminatory features within the original signal. Using wavelet coefficients directly in QSAR/QSPR modeling provides a set of low-dimensional, information-rich descriptors that capture the shape and character of the raw data distribution[129] and help to build more parsimonious models.[124]

Wavelet coefficient descriptors (WCDs)[130,131] exemplify how the WT enhances the quality of current descriptor technology and enables the development of improved models. WCDs are an adaptation of the transferable atom equivalent (TAE) descriptors developed by Breneman and Rhem.[132] TAE descriptors are derived by quantifying the distributions of multiple electronic properties computed on electronic van der Waals surfaces, which are defined as the 0.002-e-au^{-3} isosurfaces (Figure 15). TAE descriptors encode the

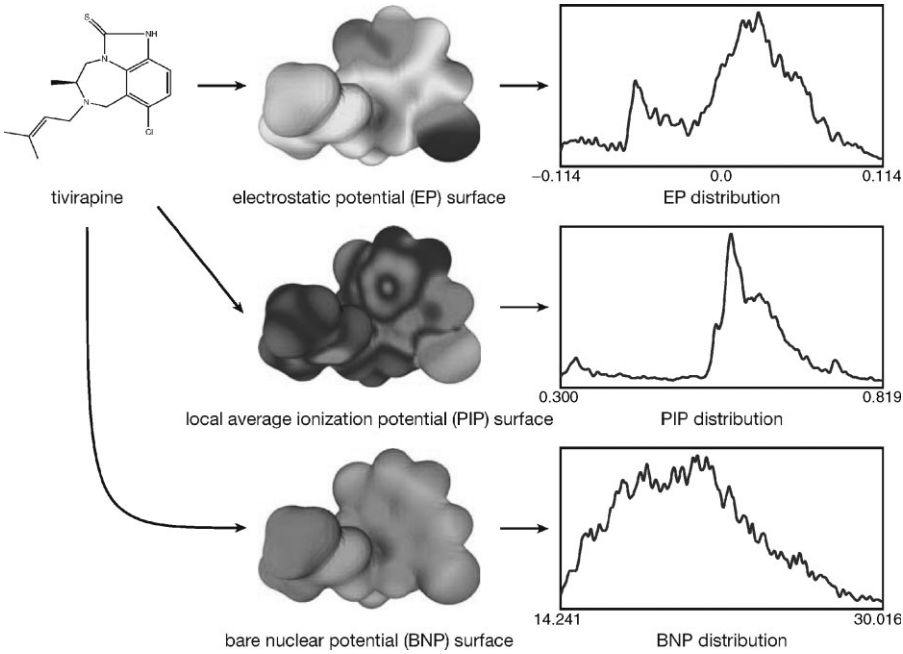


Figure 15 The HIVrt tivirapine, with its electronic van der Waals surface encoded with three electron density-derived properties and their respective property distributions.

distributions of electron density-based molecular properties such as electronic kinetic energy densities,[133] local average ionization potential,[134] electrostatic potential,[135–138] Fukui functions,[139–142] electron density gradients, and electron density Laplacian,[133] in addition to the density. The term ''TAE descriptor'' refers to a set of histograms with fixed-width bins that characterize surface property distributions (for one example, see Figure 16). Although the TAE descriptors are capable of generating high-quality models, they are nonorthogonal, in that histogram representations of property distributions contain correlated information from the same property. WCDs redefine and simplify TAE descriptor data into a stable, orthogonal representation.

Using multiresolution analysis, each property distribution is transformed into wavelet space, separating the data into frequency bands. Because the low-frequency features of the property density distributions contain most of the chemical information, a significant compression of the distributions is achieved by retaining only the wavelet coefficients in the very lowest frequency bands. These few wavelets are sufficient to describe the overall shape and character of the property distributions (see Figure 16) without carrying redundant information about their features. WCD descriptors have been shown to improve QSAR modeling because they are inherently an orthogonal representation that separate and isolate features of each surface property distribution.

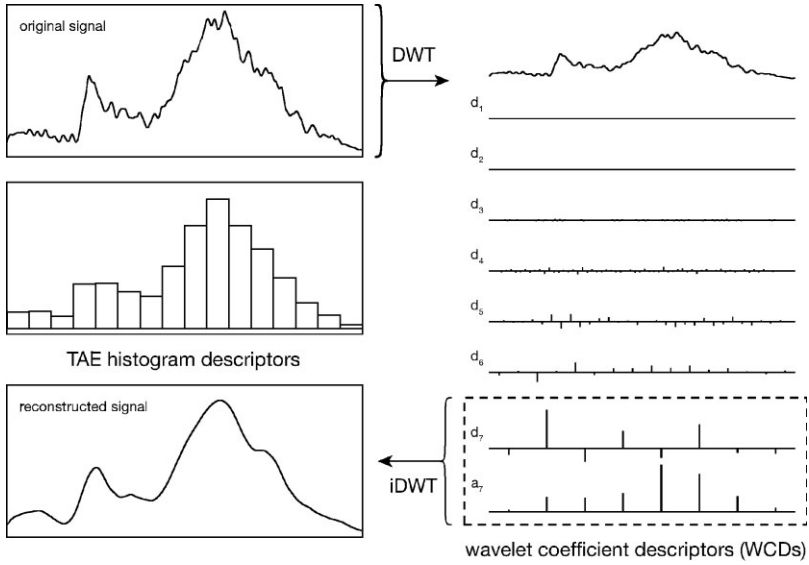


Figure 16 WCDs are generated as illustrated for each electron density-derived property. The property distribution is deconstructed using the DWT (pyramid algorithm), allowing the isolation of the lowest frequency and coarsest approximation coefficients (a_7 and d_7). These few coefficients are sufficient to reconstruct most of the original signal (via the inverse DWT) and contain the vital molecular property information needed for modeling. The WCDs replace the original TAE histogram descriptors and are orthogonal, consistent, and representative.

This aspect of WCDs ensures the resulting models to be more robust, minimizing the risk of finding spurious relationships within the set of descriptors, thereby improving overall model parsimony, and leading to greater generalizability than when using numerous, nonorthogonal descriptors.

A demonstration of the performance of WCDs in a pharmaceutical setting can be illustrated by the development of a genetic algorithm/partial least-squares (GA/PLS)[143–145] model of HIV reverse-transcriptase (HIVrt) inhibition.[146] In this example, a set of 64 molecules with assay (EC_{50}) values was used to train and evaluate a QSAR model using either TAE or WCD descriptors. Because the electronic surface properties being represented by each method are the same, this example serves to compare the benefits of the wavelet representation over the TAE surface histogram representation of these properties. The results shown in Figure 17 are taken from the cross-validated GA/PLS model predictions. It is significant to note that the number of descriptors required to produce the TAE-based model was nearly twice that of the WCD model. With its smaller number of features, the WCD-based model would be expected to be more stable and robust. It was, in fact, demonstrated during the process of model building, where WCD-based models were found

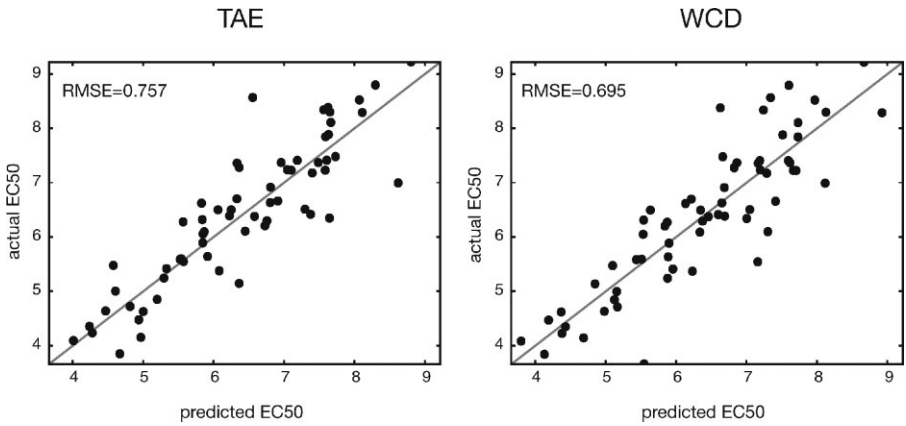


Figure 17 Comparison of HIVrt EC_{50} model parsimony using TAE and WCD descriptors. In each case, cross-validated GA/PLS models were constructed for a set of 64 HIVrt inhibitors using five PLS dimensions. The TAE model required 13 descriptors, whereas the WCD model needed only seven descriptors for similar performance.

to be less sensitive to GA/PLS tuning parameters than models built using TAE descriptors.

Other chemistry-related modeling applications have also observed improvement through the use of wavelet coefficient representations of molecular or spectral properties. An extensive list of examples of this phenomenon is available in the literature. For instance, WT methods were shown to improve spectral classification models,[147] ANN-based chromatography methods[148] and flame ionization data.[149] Although other descriptor-based methods of representing infrared spectral features have been used with some success in this application,[150–152] WT methods were found to be superior for building classification or QSAR regression models. Pattern recognition and regression using wavelet coefficients were also found to improve other infrared spectral analysis studies.[122,125,126,128,153] Because of their ability to concentrate chemical information and reduce unwanted features, a rich literature is developing around the use of wavelets as descriptors of chemical data.

Examples of the improvement possible in classification and regression analysis on experimental data using WT methods can be found in several areas: Finite impulse response (FIR) models built on impulse response improved dramatically, for example, when the signals were significantly downsampled by converting data with WTs.[154] Classifications by ANN of HPLC data of trace organic impurities[155] and thermally modulated sensor signals for various gas types[156] were improved dramatically by WT preprocessing over previous approaches.[157] The WNN[95] has been used for predicting retention times in programmed-temperature gas chromatography (PTGC)[158] and to build more generalizable models for predicting association constant values

(K_a) for benzene derivatives.[159] An interesting application of WT methods to improve the ANN classification of chromatography data is presented in the work of Yiyu, et al.,[160] where the WT was used to decompose the chromatography data and fractal analysis was used to analyze the wavelet components. The ANN categorized the compound using the "chromatographic fingerprint" or fractal dimension of the wavelet coefficients. Other studies that used WT methods to generate data descriptors giving enhanced modeling results are found in Refs. 120, and 161–163.

SUMMARY

We have illustrated the utility of WTs throughout this chapter, for cleaning, smoothing, and denoising data, as well as the benefits of their direct use as molecular property descriptors. It is clear from the examples cited that this versatile technology can identify and quantify important features within spectra or property distributions of chemical interest for use in both classification and regression models, to achieve near-linear scaling in electronic structure calculations and serve to control the quality of a basis set in ab initio molecular dynamics simulations. The evolution of wavelets in chemistry parallels the development of ever more sophisticated computational methods and hardware performance. In a relatively short period of time, wavelet methods have grown in importance from a noise filter and baseline correction tool to a fundamental component of modern data analysis, computational chemistry, and knowledge discovery.

REFERENCES

1. G. Bachman, L. Narici, and E. Beckenstein, *Fourier and Wavelet Analysis*, Springer-Verlag, New York, 2000.
2. A. Teolis, *Computational Signal Processing with Wavelets*, Birkhäuser, Boston, Massachusetts, 1998.
3. I. Daubechies, *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pennsylvania, 1992.
4. R. M. Rao and A. S. Bopardikar, *Wavelet Transforms: Introduction to Theory and Applications*, Addison-Wesley, Reading, Massachusetts, 1998.
5. A. Grossman and J. Morlet, *SIAM J. Math. Anal.*, **15**(4), 723–736 (1984). Decompositions of Hardy Functions into Square Integrable Wavelets of Constant Shape.
6. Y. Meyer, *Wavelets: Algorithms and Applications*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pennsylvania, 1993.
7. Y. Mallet, *IEEE T. Pattern Anal.*, **11**, 674–693 (1989). A Theory for Multiresolution Signal Decomposition - the Wavelet Representation.
8. R. R. Coifman, Y. Meyer, S. Quake and M. V. Wickerhauser, in *Wavelets and Their Applications*, Vol. 442, J. S. Byrnes, J. L. Byrnes, K. A. Hargreaves, and K. Berry, Eds.,

Kluwer Academic Publishers, Norwell, Massachusetts, 1994, pp. 363–379. Signal Processing and Compression with Wavelet Packets.

9. B. Walczak, *Wavelets in Chemistry*, Vol. 22, Elsevier Science, New York, 2000.
10. A. Felinger, *Data Analysis and Signal Processing in Chromatography*, Vol. 21, Elsevier Health Sciences, Amsterdam, the Netherlands, 1998.
11. B. K. Alsberg, A. M. Woodward, and D. B. Kell, *Chemom. Intell. Lab. Syst.*, **37**, 215–239 (1997). An Introduction to Wavelet Transforms for Chemometricians: A Time-Frequency Approach – A Tutorial.
12. U. Depczynski, K. Jetter, K. Molt, and A. Niemöller, *Chemom. Intell. Lab. Syst.*, **39**, 19–27 (1997). The Fast Wavelet Transform on Compact Intervals as a Tool in Chemometrics. I. Mathematical Background.
13. J. J. Workman, D. J. Veltkamp, S. Doherty, B. B. Anderson, K. E. Creasy, M. Koch, J. F. Tatera, A. L. Robinson, L. Bond, L. W. Burgess, G. N. Bokerman, A. H. Ullman, G. P. Darsey, F. Mozayeni, J. A. Bamberger, and M. S. Greenwood, *Anal. Chem.*, **71**, 121R–180R (1999). Process Analytical Chemistry.
14. X. Shao, A. K.-M. Leung, and F.-T. Chau, *Acc. Chem. Res.*, **36**, 276–283 (2003). Wavelet: A New Trend in Chemistry.
15. A. K.-M. Leung, F.-T. Chau, and J.-B. Gao, *Chemom. Intell. Lab. Syst.*, **43**, 165–184 (1998). A Review on Applications of Wavelet Transform Techniques in Chemical Analysis: 1989–1997.
16. S. Wold and M. Sjöström, *Chemom. Intell. Lab. Syst.*, **44**, 3–14 (1998). Chemometrics, Present and Future Success.
17. H. C. Smit and E. J. van den Heuvel, in *Chemometrics and Species Identification*, Vol. 141, C. Armanino, M. Forina, P. H. E. Gardiner, E. J. Van Den Heuvel, G. Kateman, S. Lanteri, H. C. Smit, and B. G. M. Vandeginste, Eds., Springer-Verlag, New York, 1987, pp. 63–89. Signal and Data Analysis in Chromatography.
18. J. Trygg, Wavelets in Chemometrics – Compression, Denoising, and Feature Extraction, 2003 Available: *http://www.acc.umu.se/~tnkjtg/Chemometrics/Editorial*.
19. B. B. Hubbard, *The World According to Wavelets*, A K Peters, Ltd., Natick, Massachusetts, 1998.
20. C. Gonnet and B. Torrésani, *Signal Process.*, **37**, 389–404 (1994). Local Frequency Analysis with the Two-Dimensional Wavelet Transform.
21. J. Blanc-Talon and D. C. Popescu, *Imaging and Vision Systems: Theory, Assessment and Applications*, Nova Science Publishers, Inc., Hauppauge, New York, 2001.
22. M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi *Wavelet Toolbox User's Guide Version 3*, The MathWorks, Inc., Natick, Massachusetts, 2004.
23. B. Walczak and D. L. Massart, *Chemom. Intell. Lab. Syst.*, **36**, 81–94 (1997). Noise Suppression and Signal Compression Using the Wavelet Packet Transform – A Tutorial.
24. R. R. Coifman and M. V. Wickerhauser, *IEEE T. Inform. Theory*, **38**(2), 713–718 (1992). Entropy-Based Algorithms for Best Basis Selection.
25. V. J. Barclay and R. F. Bonner, *Anal. Chem.*, **69**, 78–90 (1997). Application of Wavelet Transforms to Experimental Spectra: Smoothing, Denoising, and Data Set Compression.
26. C. Cai and P. d. B. Harrington, *J. Chem. Inf. Comput. Sci.*, **38**, 1161–1170 (1998). Different Discrete Wavelet Transforms Applied to Denoising Analytical Data.
27. C. R. Mittermayr, S. G. Nikolov, H. Hutter, and M. Grasserbauer, *Chemom. Intell. Lab. Syst.*, **34**, 187–202 (1996). Wavelet Denoising of Gaussian Peaks: A Comparative Study.
28. C. Perrin, B. Walczak, and D. L. Massart, *Anal. Chem.*, **73**, 4903–4917 (2001). The Use of Wavelets for Signal Denoising in Capillary Electrophoresis.
29. D. L. Donoho, in *Different Perspectives on Wavelets*, Vol. 47, I. Daubechies, Ed., American Mathematical Society, Providence, Rhode Island, 1993, pp. 173–205. Nonlinear Wavelet Methods for Recovery of Signals, Densities, and Spectra from Indirect and Noisy Data.

30. M. Wolkenstein, H. Hutter, and M. Grasserbauer, *Fresenius J. Anal. Chem.*, **358**, 165–169 (1997). Wavelet Filtering for Analytical Data.

31. D. L. Donoho, *IEEE T. Inform. Theory*, **41**(3), 613–627 (1995). De-Noising by Soft-Thresholding.

32. S. Sakakibara, in *Wavelets: Theory, Algorithms, and Applications*, C. K. Chui, L. Montefusco, and L. Puccio, Eds., Academic Press, New York, 1994, pp. 179–196. A Practice of Data Smoothing by B-spline Wavelets.

33. B. K. Alsberg, A. M. Woodward, M. K. Winson, J. Rowland, and D. B. Kell, *Analyst*, **122**, 645–652 (1997). Wavelet Denoising of Infrared Spectra.

34. C. R. Mittermayr, H. Frischenschlager, E. Rosenberg, and M. Grasserbauer, *Fresenius J. Anal. Chem.*, **358**, 456–464 (1997). Filtering and Integration of Chromatographic Data: A Tool to Improve Calibration?

35. X. Shao and W. Cai, *J. Chemom.*, **12**, 85–93 (1998). Resolution of Multicomponent Chromatograms by Window Factor Analysis with Wavelet Transform Preprocessing.

36. D. Jouan-Rimbaud, B. Walczak, R. J. Poppi, O. E. de Noord, and D. L. Massart, *Anal. Chem.*, **69**, 4317–4323 (1997). Application of Wavelet Transform To Extract the Relevant Component from Spectral Data for Multivariate Calibration.

37. A. M. Woodward, B. K. Alsberg, and D. B. Kell, *Chemom. Intell. Lab. Syst.*, **40**, 101–107 (1998). The Effect of Heteroscedastic Noise on the Chemometric Modelling of Frequency Domain Data.

38. H.-W. Tan and S. D. Brown, *J. Chemom.*, **16**, 228–240 (2002). Wavelet Analysis Applied to Removing Non-Constant, Varying Spectroscopic Background in Multivariate Calibration.

39. X.-Q. Lu and J.-Y. Mo, *Analyst*, **121**, 1019–1024 (1996). Spline Wavelet Multi-Resolution Analysis for High-Noise Digital Signal Processing in Ultraviolet-Visible Spectrophotometry.

40. M. Wolkenstein, T. Stubbings, and H. Hutter, *Fresenius J. Anal. Chem.*, **365**, 63–69 (1999). Robust Automated Three-Dimensional Segmentation of Secondary Ion Mass Spectrometry Image Sets.

41. M. Wolkenstein, H. Hutter, S. G. Nikolov, and M. Grasserbauer, *Fresenius J. Anal. Chem.*, **357**, 783–788 (1997). Improvement of SIMS Image Classification by Means of Wavelet De-Noising.

42. S. G. Nikolov, H. Hutter, and M. Grasserbauer, *Chemom. Intell. Lab. Syst.*, **34**, 263–273 (1996). De-Noising of SIMS Images via Wavelet Shrinkage.

43. X.-N. Li, Y.-Z. Liang, and F.-T. Chau, *Chemom. Intell. Lab. Syst.*, **63**, 139–153 (2002). Smoothing Methods Applied to Dealing with Heteroscedastic Noise in GC/MS.

44. H. Fang and H.-Y. Chen, *Anal. Chim. Acta*, **346**, 319–325 (1997). Wavelet Analyses of Electroanalytical Chemistry Responses and an Adaptive Wavelet Filter.

45. L. Bao, J. Mo, and Z. Tang, *Anal. Chem.*, **69**, 3053–3057 (1997). The Application in Processing Analytical Chemistry Signals of a Cardinal Spline Approach to Wavelets.

46. D. N. S. Permann and H. Teitelbaum, *J. Phys. Chem.*, **97**, 12670–12673 (1993). Wavelet Fast Fourier Transform (WFFT) Analysis of a Millivolt Signal for a Transient Oscillating Chemical Reaction.

47. B. B. Goldman and W. T. Wipke, *J. Chem. Inf. Comput. Sci.*, **40**, 644–658 (2000). Quadratic Shape Descriptors. 1. Rapid Superposition of Dissimilar Molecules Using Geometrically Invariant Surface Descriptors.

48. J. Mao, P. Sun, Z. Pan, Q. Su, and Z. Maosen, *Fresenius J. Anal. Chem.*, **361**, 140–142 (1998). Wavelet Analysis on Photoacoustic Spectra of Degraded PVC.

49. G. Beylkin, *SIAM J. Numer. Anal.*, **29**(6), 1716–1740 (1992). On the Representation of Operators in Bases of Compactly Supported Wavelets.

50. W. Dahmen and C. A. Micchelli, *SIAM J. Numer. Anal.*, **30**(2) 507–537 (1993). Using the Refinement Equation for Evaluating Integrals of Wavelets.

51. E. J. Stollnitz, T. D. Derose, and D. H. Salesin, *IEEE Comput. Graph.*, **15**(3), 76–84 (1995). Wavelets for Computer Graphics: A Primer, Part 1 – A Tutorial.
52. E. J. Stollnitz, T. D. Derose, and D. H. Salesin, *IEEE Comput. Graph.*, **15**(4), 75–85 (1995). Wavelets for Computer Graphics: A Primer, Part 2 - A Tutorial.
53. D. Marr and E. Hildreth, *Proc. R. Soc. Lond. B. Bio.*, **207**(1167), 187–217 (1980). Theory of Edge Detection.
54. J.-P. Antoine, R. Murenzi, and B. Piette, in *Wavelet and Applications*, Vol. 20, Y. Meyer, Ed., Masson and Springer-Verlag, Paris, 1992, pp. 144–159. Image Analysis with 2D Continuous Wavelet Transform: Detection of Position, Orientation and Visual Contrast of Simple Objects.
55. R. R. Klevecz, *Funct. Integr. Genomics*, **1**, 186–192 (2000). Dynamic Architecture of the Yeast Cell Cycle Uncovered by Wavelet Decomposition of Expression Microarray Data.
56. P. Teppola and P. Minkkinen, *J. Chemom.*, **14**, 383–399 (2000). Wavelet-PLS Regression Models for Both Exploratory Data Analysis and Process Monitoring.
57. F. Ehrentreich, *Anal. Bioanal. Chem.*, **372**, 115–121 (2002). Wavelet Transform Applications in Analytical Chemistry.
58. X. Shao, W. Cai, P. Sun, M. Zhang, and G. Zhao, *Anal. Chem.*, **69**, 1722–1725 (1997). Quantitative Determination of the Components in Overlapping Chromatographic Peaks Using Wavelet Transform.
59. L. Sun, W. Cai, and X. Shao, *Fresenius J. Anal. Chem.*, **370**, 16–21 (2001). A Two-Dimensional Immune Algorithm for Resolution of Overlapping Two-Way Chromatograms.
60. S. Hai-Lin, W. Ji-Hong, L. Yi-Zeng, and C. Wen-Can, *Chem. J. Chinese U.*, **18**(4), 530–534 (1997). Multiresolution Analysis of Hyphenated Chromatographic Data.
61. H. Shen, J. Wang, Y. Liang, K. Pettersson, M. Josefson, J. Gottfries, and F. Lee, *Chemom. Intell. Lab. Syst.*, **37**, 261–269 (1997). Chemical Rank Estimation by Multiresolution Analysis for Two-way Data in the Presence of Background.
62. X. Shao, W. Cai, and P. Sun, *Chemom. Intell. Lab. Syst.*, **43**, 147–155 (1998). Determination of the Component Number in Overlapping Multicomponent Chromatogram Using Wavelet Transform.
63. X. Shao and S. Hou, *Anal. Sci.*, **15**, 681–684 (1999). On-Line Resolution of Overlapping Chromatograms Using the Wavelet Transform.
64. X. Zhang, J. Jin, J. Zheng, and H. Gao, *Anal. Bioanal. Chem.*, **377**(7–8), 1153–1158 (2003). Genetic Algorithms Based on Wavelet Transform for Resolving Simulated Overlapped Spectra.
65. N. Lu, F. Wang, and F. Gao, *Ind. Eng. Chem. Res.*, **42**, 4198–4207 (2003). Combination Method of Principal Component and Wavelet Analysis for Multivariate Process Montoring and Fault Diagnosis.
66. C. L. Stork, D. J. Veltkamp, and B. R. Kowalski, *Appl. Spectrosc.*, **52**(10), 1348–1352 (1998). Detecting and Identifying Spectral Anomalies Using Wavelet Processing.
67. L. Nie, S. Wu, X. Lin, L. Zheng, and L. Rui, *J. Chem. Inf. Comput. Sci.*, **42**, 274–283 (2002). Approximate Derivative Calculated by Using Continuous Wavelet Transform.
68. J. Chen and X. Z. Wang, *J. Chem. Inf. Comput. Sci.*, **41**, 992–1001 (2001). A New Approach to Near-Infrared Spectral Data Analysis Using Independent Component Analysis.
69. D. Broadhurst, R. Goodacre, A. Jones, J. J. Rowland, and D. B. Kell, *Anal. Chim. Acta*, **348**, 71–86 (1997). Genetic Algorithms as a Method for Variable Selection in Multiple Linear Regression and Partial Least Squares Regression, with Applications to Pyrolysis Mass Spectrometry.
70. H. Martens, M. Høy, B. M. Wise, R. Bro, and P. B. Brockhoff, *J. Chemom.*, **17**, 153–165 (2003). Pre-Whitening of Data by Covariance-Weighted Pre-Processing.
71. J. A. Hageman, M. Streppel, R. Wehrens, and L. M. C. Buydens, *J. Chemom.*, **17**, 427–437 (2003). Wavelength Selection with Tabu Search.

72. J. M. Brenchley, U. Hörchner, and J. H. Kalivas, *Appl. Spectrosc.*, **51**(5), 689–699 (1997). Wavelength Selection Characterization for NIR Spectra.

73. S. L. Shew, Method and Apparatus for Determining Relative Ion Abundances in Mass Spectrometry Utilizing Wavelet Transforms. Patent #5,436,447, 1995.

74. X. Shao, L. Shao, and G. Zhao, *Anal. Commun.*, **35**, 135–137 (1998). Extraction of Extended X-ray Absorption Fine Structure Information from the Experimental Data Using the Wavelet Transform.

75. Y. Ding, T. Nanba, and Y. Miura, *Phys. Rev. B: Condens. Matter*, **58**(21), 14279–14287 (1998). Wavelet Analysis of X-Ray Diffraction Pattern for Glass Structures.

76. D. Barache, J.-P. Antoine, and J. M. Dereppe, *J. Magn. Reson.*, **128**, 1–11 (1997). The Continuous Wavelet Transform, An Analysis Tool for NMR Spectroscopy.

77. P. Guillemain, R. Kronland-Martinet, and B. Martens, in *Wavelet and Applications*, Vol. 20, Y. Meyer, Ed., Masson and Springer-Verlag, Paris, 1992, pp. 38–60. Estimation of Spectral Lines with the Help of the Wavelet Transform – Applications in NMR Spectroscopy.

78. G. Neue, *Solid State Nucl. Magn. Reson.*, **5**, 305–314 (1996). Simplification of Dynamic NMR Spectroscopy by Wavelet Transforms.

79. F. G. Meyer and G. McCarthy, in *Information Processing in Medical Imaging: 17th International Conference, IPMI 2001, Davis, California, USA, June 18–22, 2001, Proceedings*, M. F. Insana and R. M. Leahy, Eds. Springer-Verlag, New York, 2001, pp. 232–238. Estimation of Baseline Drifts in fMRI.

80. H. Serrai, L. Senhadji, J. D. de Certaines, and J. L. Coatrieux, *J. Magn. Reson.*, **124**, 20–34 (1997). Time-Domain Quantification of Amplitude, Chemical Shift, Apparent Relaxation Time T*2, and Phase by Wavelet-Transform Analysis. Application to Biomedical Magnetic Resonance Spectroscopy.

81. U. L. Günther, C. Ludwig, and H. Rüterjans, *J. Magn. Reson.*, **156**, 19–25 (2002). WAVEWAT - Improved Solvent Suppression in NMR Spectra Employing Wavelet Transforms.

82. V. L. Martins, L. F. de Almeida, S. L. de Castro, R. K. H. Galvão, M. C. U. de Araújo, and E. C. da Silva, *J. Chem. Inf. Comput. Sci.*, **43**(6), 1725–1732 (2003). A Multiscale Wavelet Data Treatment for Reliable Localization of Inflection Points for Analytical Purposes.

83. R. A. DeVore, B. Jawerth, and B. J. Lucier, *IEEE T. Inform. Theory*, **38**(2), 719–746 (1992). Image Compression Through Wavelet Transform Coding.

84. J. Le Moigne, Multi-Sensor Image Registration, Fusion and Dimension Reduction, 2005 Available: http://satjournal.tcom.ohiou.edu/issue03/applications.html.

85. J. Le Moigne and I. Zavorin, in *Use of Wavelets for Image Registration*, Wavelet Applications VII, SPIE Aerosense 2000, Orlando, Florida, 2000.

86. J. M. Shapiro, *IEEE T. Signal Proces.*, **41**(12), 3445–3462 (1993). Embedded Image Coding Using Zerotrees of Wavelet Coefficients.

87. L. J. Chipman, T. M. Orr, and L. N. Graham, in *Wavelet Applications in Signal and Image Processing III, SPIE*, July 12–14, 1995, San Diego, California, *Proceedings*, A. F. Laine, M. A. Unser, and M. V. Wickerhauser, Eds., SPIE, the International Society for Optical Engineering, Bellingham, Massachusetts, 1995, pp. 208–219. Wavelets and Image Fusion.

88. F. Peyrin, M. Zaim, and R. Goutte, *J. Math. Imaging Vis.*, **3**, 105–121 (1993). Construction of Wavelet Decompositions for Tomographic Images.

89. H. Li, S. Manjunath, and S. K. Mitra, *Graph. Model. Im. Proc.*, **57**(3), 235–245 (1995). Multisensor Image Fusion Using the Wavelet Transform.

90. Z. Zhang and R. S. Blum, in *Multisensor Image Fusion Using a Region-Based Wavelet Transform Approach*, DARPA Image Understanding Workshop 1997, New Orleans, Louisiana, 1997; pp. 1447–1451.

91. A. K.-M. Leung, F.-T. Chau, J.-B. Gao, and T.-M. Shih, *Chemom. Intell. Lab. Syst.*, **43**, 69–88 (1998). Application of Wavelet Transform in Infrared Spectrometry: Spectral Compression and Library Search.

92. B. Walczak and D. L. Massart, *Chemom. Intell. Lab. Syst.*, **38**, 39–50 (1997). Wavelet Packet Transform Applied to a Set of Signals: A New Approach to the Best-Basis Selection.

93. K.-K. Li, F.-T. Chau, and A. K.-M. Leung, *Chemom. Intell. Lab. Syst.*, **52**, 135–143 (2000). Compression of Ultraviolet-Visible Spectrum with Recurrent Neural Network.

94. F.-T. Chau, J.-B. Gao, T.-M. Shih, and J. Wang, *Appl. Spectrosc.*, **51**(5), 649–659 (1997). Compression of Infrared Spectral Data Using the Fast Wavelet Transform Method.

95. L. Wei and L. Jinping, *Chin. Sci. Bull.*, **42**(10), 822–826 (1997). The Compression of IR Spectra by Using Wavelet Neural Network.

96. K. L. Peterson, in *Reviews in Computational Chemistry*, Vol. 16, K. B. Lipkowitz and D. B. Boyd, Eds., Wiley-VCH, New York, 2000, pp. 53–140. Artificial Neural Networks and Their Use in Chemistry.

97. K. J. Cho, T. A. Arias, J. D. Joannopoulos, and P. K. Lam, *Phys. Rev. Lett.*, **71**(12), 1808–1811 (1993). Wavelets in Electronic Structure Calculations.

98. M. F. Herman, *Annu. Rev. Phys. Chem.*, **45**, 83–111 (1994). Dynamics by Semiclassical Methods.

99. R. Kosloff, *Annu. Rev. Phys. Chem.*, **45**, 145–178 (1994). Propagation Methods for Quantum Molecular Dynamics.

100. H. Nakamura, *Annu. Rev. Phys. Chem.*, **48**, 299–328 (1997). Theoretical Studies of Chemical Dunamics: Overview of Some Functional Mechanisms.

101. T. A. Arias, *Rev. Mod. Phys.*, **71**(1), 267–311 (1999). Multiresolution Analysis of Electronic Structure: Semicardinal and Wavelet Bases.

102. D. Feller and E. R. Davidson, in *Reviews in Computational Chemistry*, Vol. 1, K. B. Lipkowitz and D. B. Boyd, Eds., VCH Publishers, Weinheim, Germany, 1990, pp. 1–43. Basis Sets for Ab Initio Molecular Orbital Calculations and Intermolecular Interactions.

103. T. A. Arias, K. J. Cho, J. D. Joannopoulos, P. K. Lam, and M. P. Teter, in *Toward Teraflop Computing and New Grand Challenge Applications*, R. K. Kalia and P. Vashishta, Eds., Nova Science Publishers, Inc., Hauppauge, New York, 1995, pp. 25–36. Wavelet-Transform Representation of the Electronic Structure of Materials.

104. S. Wei and M. Y. Chou, *Phys. Rev. Lett.*, **76**(15), 2650–2653 (1996). Wavelets in Self-Consistent Electronic Structure Calculations.

105. P. Fischer and M. Defranceschi, in *Conceptual Trends in Quantum Chemistry*, E. S. Kryachko and J. L. Calais, Eds., Kluwer Academic, Dordrecht, the Netherlands, 1994, pp. 227–247. The Wavelet Transform: A New Mathematical Tool for Quantum Chemistry.

106. S. Nagy and P. János, *Int. J. Quantum Chem.*, **84**, 523–529 (2001). Multiresolution Analysis of Density Operators, Electron Density, and Energy Functionals.

107. S. S. Iyengar and M. J. Frisch, *J. Chem. Phys.*, **121**(11), 5061–5070 (2004). Effect of Time-Dependent Basis Functions and Their Superposition Error on Atom-Centered Density Matrix Propagation (ADMP): Connections to Wavelet Theory of Multiresolution Analysis.

108. N. R. Kestner and J. E. Combariza, in *Reviews in Computational Chemistry*, Vol. 13, K. B. Lipkowitz and D. B. Boyd, Eds., Wiley-VCH, New York, 1999, pp. 99–132. Basis Set Superposition Errors: Theory and Practice.

109. J. P. Modisette, P. Nordlander, J. L. Kinsey, and B. R. Johnson, *Chem. Phys. Lett.*, **250**, 485–494 (1996). Wavelet Bases in Eigenvalue Problems in Quantum Mechanics.

110. P. Fischer and M. Defranceschi, *Int. J. Quantum Chem.*, **45**, 619–636 (1993). Looking at Atomic Orbitals Through Fourier and Wavelet Transforms.

111. P. Fischer and M. Defranceschi, in *Wavelets: Theory, Algorithms and Applications*, C. K. Chui, L. Montefusco, and L. Puccio, Eds., Academic Press, New York, 1994, pp. 495–506. Representation of the Atomic Hartree-Fock Equations in a Wavelet Basis by Means of the BCR Algorithm.

112. E. von Kitzing and E. Schmitt, *THEOCHEM*, **336**, 245–259 (1995). Configurational Space of Biological Macromolecules as Seen by Semi-empirical Force Fields: Inherent Problems for Molecular Design and Strategies to Solve Them by Means of Hierarchical Force Fields.

113. R. J. Harrison, G. I. Fann, T. Yanai, Z. Gan, and G. Beylkin, *J. Chem. Phys.*, **121**(23), 11587–11598 (2004). Multiresolution Quantum Chemistry: Basic Theory and Initial Applications.

114. T. Yanai, G. I. Fann, Z. Gan, R. J. Harrison, and G. Beylkin, *J. Chem. Phys.*, **121**(14), 6680–6688 (2004). Multiresolution Quantum Chemistry in Multiwavelet Bases: Hartree-Fock Exchange.

115. T. Yanai, G. I. Fann, Z. Gan, R. J. Harrison, and G. Beylkin, *J. Chem. Phys.*, **121**(7), 2866–2876 (2004). Multiresolution Quantum Chemistry in Multiwavelet Bases: Analytic Derivatives for Hatree-Fock and Density Functional Theory.

116. L. B. Kier and L. H. Hall, *Molecular Connectivity In Chemistry and Drug Research*, Academic Press, New York, 1976.

117. L. B. Kier and L. H. Hall, *Molecular Connectivity in Structure-Activity Analysis*, John Wiley, London, 1986.

118. L. B. Kier, L. H. Hall, W. J. Murray, and M. Randic, *J. Pharm. Sci.*, **64**(12), 1971–1974 (1975). Molecular Connectivity. I: Relationship to Nonspecific Local Anesthesia.

119. C. Hansch and A. Leo, *Exploring QSAR*, American Chemical Society, Washington, D.C., 1995.

120. M. Cocchi, R. Seeber, and A. Ulrici, *Chemom. Intell. Lab. Syst.*, **57**, 97–119 (2001). WPTER: Wavelet Packet Transform for Efficient Pattern Recognition of Signals.

121. L. I. Nord and S. P. Jacobsson, *Chemom. Intell. Lab. Syst.*, **44**, 153–160 (1998). A Novel Method for Examination of the Variable Contribution to Computational Neural Network Models.

122. Y. Mallet, D. Coomans, J. Kautsky, and O. de Vel, *IEEE T. Pattern Anal.*, **19**(10), 1058–1066 (1997). Classification Using Adaptive Wavelets for Feature Extraction.

123. P. Wunsch and A. F. Laine, *Pattern Recogn.*, **28**(8), 1249 (1995). Wavelet Descriptors for Multiresolution Recognition of Handprinted Characters.

124. O. E. d. Noord, *Chemom. Intell. Lab. Syst.*, **23**, 65–70 (1994). The Influence of Data Preprocessing on the Robustness and Parsimony of Multivariate Calibration Models.

125. Y. Mallet, D. Coomans, and O. de Vel, *Chemom. Intell. Lab. Syst.*, **35**, 157–173 (1996). Recent Development in Discriminant Analysis on High Dimensional Spectral Data.

126. M. Bos and J. A. M. Vrielink, *Chemom. Intell. Lab. Syst.*, **23**, 115–122 (1994). The Wavelet Transform for Pre-processing IR Spectra in the Identification of Mono- and Di-substituted Benzenes.

127. B. Walczak, E. Bouveresse, and D. L. Massart, *Chemom. Intell. Lab. Syst.*, **36**, 41–51 (1997). Standardization of Near-Infrared Spectra in the Wavelet Domain.

128. B. Walczak, B. van den Bogaert, and D. L. Massart, *Anal. Chem.*, **68**, 1742–1747 (1996). Application of Wavelet Packet Transform in Pattern Recognition of Near-IR Data.

129. P. D. B. Harrington, A. Urbas, and C. Wan, *Anal. Chem.*, **72**, 5004–5013 (2000). Evaluation of Neural Network Models with Generalized Sensitivity Analysis.

130. C. M. Breneman, N. Sukumar, K. P. Bennett, M. J. Embrechts, C. M. Sundling, and L. Lockwood, in *QSAR In Vivo: American Chemical Society National Meeting*, American Chemical Society, Washington, D.C., 2000. Wavelet Representations of Molecular Electronic Properties: Applications in ADME, QSPR, and QSAR.

131. L. Lockwood. 1. Effects of a Trimethylsilyl Substituent on the Photochemical Rearrangement of 2,5-cyclohexadienones. 2. Development and Implementation of Novel Electron Density Derived Molecular Surface Descriptors. Ph.D. dissertation, Rensselaer Polytechnic Institute, Troy, New York, 2002.

132. C. M. Breneman and M. Rhem, *J. Comput. Chem.*, **18**(2), 182–197 (1997). A QSPR Analysis of HPLC Column Capacity Factors for a set of High-Energy Materials Using Electronic Van der Waals Surface Property Descriptors Computed by the Transferable Atom Equivalent Method.

133. R. F. W. Bader, *Atoms in Molecules: A Quantum Theory*, Oxford Press, Oxford, United Kingdom, 1990.

134. J. S. Murray and P. Politzer, in *Theoretical Organic Chemistry*, Vol. 5, C. Párkányi, Ed., Elsevier Science Publishers BV, Amsterdam, the Netherlands, 1998, pp. 189–202. Average Local Ionization Energies: Significance and Applications.

135. P. Politzer and D. G. Truhlar, *Chemical Applications of Atomic and Molecular Electrostatic Potential*, Plenum Press, New York, 1981.

136. P. Politzer, N. Sukumar, K. Jayasuriya, and S. Ranganathan, *J. Am. Chem. Soc.*, **110**, 3425–3430 (1988). A Computational Evaluation and Comparison of Nitramine Properties.

137. J. S. Murray, N. Sukumar, S. Ranganathan, and P. Politzer, *Int. J. Quantum Chem.*, **37**(5), 611–629 (1990). A Computational Analysis of the Electrostatic Potentials and Relative Bond Strengths of Hydrazine and Some of its 1,1-Dimethyl Derivatives.

138. C. M. Breneman and M. Martinov, in *Molecular Electrostatic Potentials: Concepts and Applications*, Vol. 3, J. S. Murray and K. Sen, Eds., Elsevier, Amsterdam, the Netherlands, 1996, pp. 143–179. The Use of Electrostatic Potential Fields in QSAR and QSPR.

139. K. Fukui, T. Yonezawa, and C. Nagata, *J. Chem. Phys.*, **27**(6), 1247–1259 (1957). MO-Theoretical Approach to the Mechanism of Charge Transfer in the Process of Aromatic Substitutions.

140. K. Fukui, *Theory of Orientation and Stereoselection*, Springer-Verlag, Berlin, Germany, 1975.

141. R. G. Parr and W. Yang, *J. Am. Chem. Soc.*, **106**, 4049–4050 (1984). Density-Functional Approach to the Frontier-Electron Theory of Chemical Rectivity.

142. R. G. Parr and W. Yang, *Density-Functional Theory of Atoms and Molecules*, Oxford University Press, New York, 1989.

143. K. Hasegawa and K. Funatsu, *SAR QSAR Environ. Res.*, **11**(3–4), 189–209 (2000). Partial Least-Squares Modeling and Genetic Algorithm Optimization in Quantitative Structure-Activity Relationships.

144. K. Hasegawa, Y. Miyashita, and K. Funatsu, *J. Chem. Inf. Comput. Sci.*, **37**(2), 306–310 (1997). GA Strategy for Variable Selection in QSAR Studies: GA-Based PLS Analysis of Calcium Channel Antagonists.

145. M. Ozdemir, Evolutionary Computing for Feature Selection and Predictive Data Mining. Ph.D. dissertation, Rensselaer Polytechnic Institute, Troy, New York, 2002.

146. R. Garg, S. P. Gupta, H. Gao, M. S. Babu, A. K. Debnath, and C. Hansch, *Chem. Rev.*, **99**, 3525–3691 (1999). Comparative Quantitative Structure-Activity Relationship Studies on Anti-HIV Drugs.

147. B. Debska, B. Guzowska-Swider, and D. Carbol-Bass, *J. Chem. Inf. Comput. Sci.*, **40**, 330–338 (2000). Automatic Generation of Knowledge Base from Infrared Spectral Database for Substructure Recognition.

148. C. Cai and P. d. B. Harrington, *Anal. Chem.*, **71**, 4134–4141 (1999). Prediction of Substructure and Toxicity of Pesticides with Temperature Constrained-Cascade Correlation Network from Low-Resolution Mass Spectra.

149. M. Jalali-Heravi and M. H. Fatemi, *J. Chromatogr. A*, **825**, 161–169 (1998). Prediction of Flame Ionization Detector Response Factors Using an Artificial Neural Network.

150. C. M. R. Ginn, D. B. Turner, and P. Willett, *J. Chem. Inf. Comput. Sci.*, **37**, 23–37 (1997). Similarity Searching in Files of Three-Dimensional Chemical Structures: Evaluation of the EVA Descriptor and Combination of Rankings Using Data Fusion.

151. A. M. Ferguson, T. W. Heritage, P. Jonathon, S. E. Pack, L. Phillips, J. Rogan, and P. J. Snaith, *J. Comput. Aided Mol. Des.*, **11**, 143–152 (1997). EVA: A New Theoretically Based Molecular Descriptor for Use in QSAR/QSPR Analysis.

152. T. W. Heritage, A. M. Ferguson, D. B. Turner, and P. Willett, *Perspect. Drug Discov.*, **9–11**, 381–398 (1998). EVA: A Novel Theoretical Descriptor for QSAR Studies.

153. J. Trygg and S. Wold, *Chemom. Intell. Lab. Syst.*, **42**, 209–220 (1998). PLS Regression on Wavelet Compressed NIR Spectra.

154. M. Nikolaou and P. Vuthandam, *AIChE J.*, **44**(1), 141–150 (1998). FIR Model Identification: Parsimony Through Kernel Compression with Wavelets.

155. E. R. Collantes, R. Duta, W. J. Welsh, W. L. Zielinski, and J. Brower, *Anal. Chem.*, **69**, 1392–1397 (1997). Preprocessing of HPLC Trace Impurity Patterns by Wavelet Packets for Pharmaceutical Fingerprinting Using Artificial Neural Networks.

156. S. Maldonado-Bascón, S. Al-Khalifa, and F. López-Ferreras, in *Computational Methods in Neural Modeling: Seventh International Conference, IWANN 2003, Mao, Spain, June 3–6, 2003, Proceedings*, Vol. 2687, J. Méra and J. R. Alvarez, Eds., Springer-Verlag, New York, 2003, pp. 798–805. Feature Reduction Using Support Vector Machines for Binary Gas Detection.

157. P. Althainz, J. Goschnick, S. Ehrmann, and H. J. Ache, *Sensor. Actuat. B*, **33**, 72–76 (1996). Multisensor Microsystem for Contaminants in Air.

158. X. Zhang, J. Qi, R. Zhang, M. Liu, Z. Hu, H. Xue, and B. Fan, *Comput. Chem.* (*Oxford*), **25**, 125–133 (2001). Prediction of Programmed-Temperature Retention Values of Naphthas by Wavelet Neural Networks.

159. L. Liu and Q.-X. Guo, *J. Chem. Inf. Comput. Sci.*, **39**, 133–138 (1999). Wavelet Neural Network and Its Application to the Inclusion of beta-Cyclodextrin with Benzene Derivatives.

160. C. Yiyu, C. Minjun, and W. J. Welsh, *J. Chem. Inf. Comput. Sci.*, **43**(6), 1959–1965 (2003). Fractal Fingerprinting of Chromatographic Profiles Based on Wavelet Analysis and Its Application to Characterize the Quality Grade of Medicinal Herbs.

161. M. Köküer, F. Murtagh, N. D. McMillan, S. Riedel, B. O'Rourke, K. Beverly, A. T. Augousti, and J. Mason, *J. Chem. Inf. Comput. Sci.*, **43**, 587–594 (2003). A Wavelet, Fourier and PCA Data Analysis Pipeline: Application to Distinguishing Mixtures of Liquids.

162. J. Zhao, X. W. Yang, J. P. Li, and Y. Y. Tang, in *Wavelet Analysis and Its Applications: Second International Conference, WAA 2001, Hong Kong, China, December 18–20, 2001, Proceedings*, Vol. 2251, Y. Y. Tang, M. V. Wickerhauser, P. C. Yuen, and C. H. Li, Eds., Springer-Verlag, New York, 2002, pp. 424–429. DNA Sequences Classification Based on Wavelet Packet Analysis.

163. G. V. Gkoutos, H. Rzepa, R. M. Clark, O. Adjei, and H. Johal, *J. Chem. Inf. Comput. Sci.*, **43**(5), 1342–1355 (2003). Chemical Machine Vision: Automated Extraction of Chemical Metadata from Raster Images.

Author Index

Abe, Y., 160
Abkevich, V. I., 228
Acharya, K. R., 160
Ache, H. J., 329
Adams, L. M., 289
Adjei, O., 329
Aggarwal, J. K., 51
Ahern, C. A., 286
Alagona, G., 224
Alard, P., 167
Alberts, B., 286
Alexandrov, N. N., 50, 51, 52
Al-Khalifa, S., 329
Allen, M. P., 166, 221, 288
Allen, T. W., 285
Allen, T., 290, 291
Alonso, D. O. V., 224, 227
Alper, H. E., 292
Alsberg, B. K., 322, 323
Althainz, P., 329
Altschul, S. F., 157, 162
Aluru, N., 292
Alvarez, J. R., 329
Amit, A. G., 157
Anderson, B. B., 322
Anderson, O. S., 285
Anderson, P. W., 220
Andricioaei, I. I., 222
Anfinsen, Jr., C. B., 49, 167, 220
Anson, M. L., 167
Antoine, J.-P., 24, 325
Apic, G., 55
Appel, A. W., 289
Appel, R. D., 157
Apweiler, R., 157, 159
Arbuckle, B. W., 292
Argos, P., 53, 159, 163
Arias, T. A., 326
Armanino, C., 322
Armen, R. S., 55
Arminski, L., 158
Armstrong, K. M., 286
Arnone, A., 160
Arteca, G. A., 54
Arthur, J. W., 161
Artymiuk, P. J., 52, 53
Arun, K. S., 52
Ashcroft, F. M., 284
Ashcroft, N. W., 292
Assa-Munt, N., 50
Aszodi, A., 48
Atilgan, A. R., 49
Attwood, T. K., 156
Augousti, A. T., 329

Babu, M. S., 328
Baccarani, G., 292
Bachar, O., 52
Bachman, G., 321
Bader, R. F. W., 327
Badman, M. K., 227
Bahar, I., 49
Bai, Y., 225
Bairoch, A., 157, 158
Bajaj, M., 161
Bakajin, O., 221
Baker, D., 49, 156, 225
Baldwin, R. L., 220
Bamberger, J. A., 322
Banavar, J. R., 55

Reviews in Computational Chemistry, Volume 22
edited by Kenny B. Lipkowitz, Thomas R. Cundari, and Valerie J. Gillet

Banks, J. L., 292
Banner, D. W., 50
Bao, L., 323
Barache, D., 325
Barclay, V. J., 322
Barker, W. C., 157, 158, 161
Barnard, J. M., 55
Barnatan, D., 54
Barret, C., 159, 164
Barthel, J. M. G., 287
Barton, G. J., 51, 53, 162
Bashford, D., 164, 224
Bassolino, D., 288
Bastug, T., 285
Bateman, A., 156, 157
Bates, P. A., 158, 164
Baumketner, A., 156, 226, 228
Baxevanis, A. D., 156
Bayley, H., 284
Bayly, C. I., 224
Bebdek, G. B., 227
Beck, D. A. C., 55
Becker, O. M., 291
Beglov, D., 291
Bellott, M., 164
Benner, S. A., 161
Bennett, K. P., 327
Berendsen, H. J. C., 166, 224, 284, 287, 288, 290, 291, 292
Berendzen, J., 161
Berg, B. A., 221
Berg, J. M., 287
Berkowitz, M. L., 287, 289
Berman, H. M., 48, 157, 158
Bernal, A., 48
Berne, B. J., 225, 291, 292
Bernèche, S., 290, 291
Bernstein, F. C., 48
Berry, R. S., 290
Betancourt, M., 228
Beverly, K., 329
Beylkin, G., 323, 327
Bezrukov, S. M.
Bhandarkar, M., 165
Bhat, T. N., 48, 157
Bigging, P. C., 286
Billeter, S. R., 224
Biltonen, R. L., 287
Binder, K., 220
Birdsall, C. K., 290
Björk, Å., 289
Blake, C. C. F., 155
Blanc-Talon, J., 322
Blankenbecler, E. R., 53
Blatter, M.-C., 157
Bloomer, A. C., 50
Blostein, S. D., 52
Blum, L., 287
Blum, R. S., 325
Blundell, T. L., 157, 158, 161, 163, 164
Board, J. A., 289
Bockris, J., 287
Boczko, E. M., 224
Boda, D., 288, 289
Boeckmann, B., 157, 158
Bohr, H., 54
Bokerman, G. N., 322
Bolhuis, P. G., 226, 227
Bond, L., 322
Bonneau, R., 156
Bonner, R. F., 322
Bopardikar, A. S., 321
Bordoli, L., 157
Bork, P., 158
Borkovec, M., 226
Bos, M., 327
Bottomley, S., 225
Bourne, P. E., 50, 53, 157, 158, 159
Bouveresse, E., 327
Bouzida, D., 222
Bowie, J. U., 53, 165
Boyd, D. B., 54, 284, 286, 287, 288, 290, 291, 326
Braha, O., 284
Branden, C., 48, 155
Brandt, A., 290
Braun, W., 164
Bray, D., 286
Brenchley, J. M., 325
Breneman, C. M., 327, 328
Brenner, S. E., 48, 50, 51
Brew, K., 155, 160
Brice, M. D., 48
Bro, R., 324
Broadhurst, D., 324
Brockhoff, P. B., 324
Brocklehurst, S. M., 163
Bromberg, S., 220
Brooks, B. R., 165, 166, 224
Brooks, C., 49
Brooks, III, C. L., 165, 220, 221, 223, 224, 225, 287
Brower, J., 329
Brown, M., 159

Brown, N. P., 48
Brown, S. D., 323
Browne, W., 155
Bruccoleri, R. E., 165, 224
Brunner, R., 165
Brutlag, D. L., 53
Bryngelson, J. D., 220
Bryson, K., 160, 164
Buchner, P., 157
Buldyrev, S. V., 223, 226
Burgess, L. W., 322
Busath, D. D., 285, 288, 290
Buydens, L. M. C., 324

Cabral, J. M., 286
Cadene, M., 286
Caflisch, A., 226, 228
Cai, C., 322, 328
Cai, M., 288
Cai, W., 323, 324
Cai, Y. D., 49
Calais, J. L., 326
Caldwell, J. W., 165, 224
Callahan, B., 223
Camacho, C. J., 222
Cantor, C. R., 48
Canutescu, A. A., 165
Carbol-Bass, D., 328
Cardenas, A. E., 225, 292
Carney, D., 164
Carré, B. A., 289
Carter, Jr., C. W., 158
Case, D. A., 165, 224
Casper, J., 159
Castro, E. D., 157
Castro-Alvear, J., 158
Catacuzzeno, L., 288
Chahine, J., 227
Chait, B. T., 286
Chan, H. S., 220, 222, 228
Chandler, D., 226, 227
Chandonia, J. M., 51
Chandrasekhar, J., 166, 290
Charge, S. B., 227
Chau, F.-T., 322, 323, 325, 326
Cheatham, III, T. E., 165
Cheley, S., 284
Chen, C., 51
Chen, D. P., 285, 288, 292
Chen, H.-Y., 323
Chen, J., 286, 324
Chen, L., 53
Chen, Y., 158, 285
Cheung, M. S., 223, 224
Chipman, L. J., 325
Chiu, S. W., 285, 287, 291, 292
Cho, K. J., 326
Chothia, C., 48, 49, 50, 55, 157, 162, 220
Chou, K. C., 49
Chou, M. Y., 326
Chu, K., 161
Chui, C. K., 323, 326
Chung, S.-H., 285, 287, 290, 291, 292, 293
Ciccarelli, F. D., 158
Ciccotti, G., 224
Cieplak, P., 224
Clamp, M., 162
Clancy, P., 292
Clark, A., 227
Clark, R. M., 329
Clarke, J., 226
Clementi, C., 221, 223, 227
Cline, M., 159
Coalson, R. D., 285, 291, 292
Coatrieux, J. L., 325
Cocchi, M., 327
Cohen, F. E., 49, 166
Cohen, M. A., 161
Cohen, S. L., 286
Coifman, R. R., 321, 322
Coin, L., 157
Collantes, E. R., 329
Colovos, C., 167
Colson, S. D., 286
Combariza, J. E., 326
Condron, M. M., 227
Conlan, S., 284
Contreras-Moreira, B., 164
Coomans, D., 327
Copley, R. R., 158
Corey, R. B., 48
Cornell, W. D., 224
Corran, P. H., 50
Corry, B., 285, 290, 293
Cortes, D. M., 286
Couch, G. S., 161
Coulson, R. M. R., 55
Coutsias, E. A., 52
Cowan, S. W., 286
Cox, J. M., 163
Cox, M. M., 156
Cramer, C. J., 225
Creasy, K. E., 322
Creighton, T. E., 48, 156

Cremer, P. S., 284
Crippen, G. M., 49, 53, 54
Croning, M. D. R., 159
Cross, T. A., 285
Crozier, P. S., 289, 290
Cuello, L. G., 286
Cuff, J., 162
Cukierman, S., 286
Cullis, A., 49
Cundari, T. R., 49, 55, 156, 293
Cunningham, P. D., 160

da Silva, E. C., 325
Daggett, V., 55, 224, 225, 226, 227
Dahlquist, G., 289
Dahmen, W., 323
Dalke, A., 49, 167, 286
Damm, W., 291
Damodaran, K. V., 286
Darden, T. A., 289
Darsey, G. P., 322
Daubechies, I., 321, 322
Daunter, Z., 161
Davidson, E. R., 284, 326
Davies, D., 48
Davis, R., 226
Davison, D. B., 156
Day, R., 55
Dayhoff, M. O., 161
de Almeida, L. F., 325
de Araújo, M. C. U., 325
de Castro, S. L., 325
de Certaines, J. D., 325
de Groot, B. L., 285
de Koening, E. J., 227
de Noord, O. E., 323
de Vel, O., 327
De Jong, D., 227
De Maeyer, M., 166
Deane, C. M., 157
Debnath, A. K., 328
DeBolt, S., 165
Debska, B., 328
Deerfield, II, D. W., 162
Defranceschi, M., 326
Delarue, M., 165
Dellago, C., 226, 227
Depczynski, U., 322
Dereppe, J. M., 325
Derose, T. D., 324
Derrida, B., 220, 221
Deserno, M., 288
Desmet, J., 166
Deuflhard, P., 284, 291
DeVore, R. A., 325
Dick, T. J., 166
Dickerson, R., 48
Diekhans, M., 159
Dietmann, S., 55
Dill, K. A., 52, 220, 221, 222, 228
Dima, R. I., 228
Ding, F., 223, 226, 227
Ding, H.-Q., 289
Ding, Y., 325
Dinner, A. R., 225, 226
Do, R. K., 163
Dobson, C. M., 221, 226, 227
Doerks, T., 158
Doherty, S., 322
Dokholyan, N. V., 223, 226, 227
Doll, J. D., 221
Domene, C., 291
Donoho, D. L., 322, 323
Doolittle, R. F., 162, 163
Dougherty, D. A., 284
Downs, G. M., 55
Doyle, D. A., 286
Draper, J., 159
Dress, A. W. M., 162
Driscoll, P. C., 54
Du, R., 222
Duan, L., 224
Dunbrack, Jr., R. L., 50, 161, 164, 165, 166
Durand-Vidal, D. P., 287
Durbin, R., 157
Duta, R., 329
Dynan, W., 156
Dyson, H. J., 225

Eastwood, J. W., 288
Eaton, W. A., 221, 226
Eck, R. V., 161
Eddy, S. R., 157, 159
Edge, C. M., 161
Edholm, O., 287
Edsall, J. T., 167
Edwards, S., 285
Efimov, A. V., 49
Eggert, D. W., 51
Ehrentreich, F., 324
Ehrlich, L. W., 289
Ehrmann, S., 329
Eisenberg, D., 53, 165, 285, 288
Eisenberg, R. S., 284, 290, 292, 293

Eisenmenger, F., 164
Elber, R., 160, 225
Elmore, D. E., 284
Elofsson, A., 160, 164
Embrechts, M. J., 327
Engel, A., 286
Engh, R. A., 167
Enright, A. J., 55
Ermak, D. L., 290
Erman, B., 49
Essmann, U., 287, 289
Estreicher, A., 157
Evanseck, J. D., 164
Ewald, P., 288

Fain, B., 54
Fan, B., 329
Fang, H., 323
Fann, G. I., 327
Faraldo-Gomez, J. D., 288
Fatemi, M. H., 328
Fedorova, N. D., 158
Feibig, K. M., 220
Feig, M., 224, 287
Felenstein, J., 162, 163
Felinger, A., 322
Feller, D., 326
Feller, S. E., 287
Feng, Z., 48, 157
Feng, Z. K., 54
Feng, Z. P., 50
Fennen, J., 224
Ferguson, A. M., 328
Ferguson, D. M., 165, 224
Fernandez-Escamilla, A. M., 224
Ferrari, C., 51
Ferrenberg, A. M., 222
Ferrin, T. E., 161
Ferro, S., 157, 158
Fersht, A. R., 220, 221, 225, 226, 227
Feytmans, E., 157
Fezoui, Y., 227
Fichtner, W., 289
Fick, A., 292
Field, M. J., 164, 166
Finke, J. M., 223
Finn, R. D., 157
Fischer, D., 52, 164
Fischer, P., 326
Fischer, S., 164
Fiser, A., 156, 163
Fisher, R. B., 51
Fitzgerald, P. M. D., 158
Flammini, A., 55
Forina, M., 322
Forrest, L. R., 287
Fox, K. R., 156
Fox, T., 224
Frantz, D. D., 221
Freeman, D. L., 221
Frenkel, D., 166
Friedel, M., 156, 228
Friedman, H. L., 290
Friesner, R. A., 166, 292
Frisch, M. J., 326
Frischenschlager, H., 323
Frishman, D., 159, 163
Fukui, K., 328
Funatsu, K., 328
Furth, A. J., 50

Galor, T., 160
Galperin, M. Y., 156
Galvão, R. K. H., 325
Gambow, H. N., 51
Gan, Z., 327
Gao, F., 324
Gao, H., 324, 328
Gao, J., 164
Gao, J.-B., 322, 325, 326
García, A. E., 222, 223, 224, 225, 227, 287
Gardiner, C. W., 292
Gardiner, E. J., 53
Gardiner, P. H. E., 322
Gardner, E., 220
Gardner, S. P., 163, 167
Garel, T., 221
Garg, R., 328
Gasteiger, E., 157, 158
Gattiker, A., 157
Geissler, P. L., 227
Gelatt, C. D., 167
Georgalis, Y., 288
Gerlt, J. A., 50
Gerstein, M., 53
Ghio, C., 224
Ghosh, A., 225
Gibas, C., 156
Gibbon, P., 284
Gibson, T. J., 162
Gibson, T., 161
Gilbert, W., 48
Gillespie, D., 288
Gillet, V., 156

Gilliland, G., 48, 157
Ginn, C. M. R., 328
Gish, W., 157, 158
Gkoutos, G. V., 329
Glosli, J., 289
Gō, N., 51, 52, 54, 164, 222
Goaman, L. C. G., 163
Goddard, III, W. A., 289
Goddard, T. D., 161
Godzik, A., 54, 160, 164, 223
Goldman, B. B., 323
Goldstein, R. F., 166
Golub, G. H., 51
Gonnet, C., 322
Gonnet, G. H., 161
Gonzalez, O., 54
Goodacre, R., 324
Goodnick, S., 284
Gordon, D. B., 166
Gore, M., 225
Gorse, D., 228
Goryll, M., 284
Goschnick, J., 329
Gottfries, J., 324
Gough, G., 55
Gould, I. R., 224
Goutte, R., 325
Gouy, M., 162
Graf, P., 285
Graham, L. N., 325
Grasserbauer, M., 322, 323
Grate, L., 159
Gray, P., 292
Greenblatt, D. M., 161
Greengard, L., 288
Greenwood, M. S., 322
Greer, J., 158
Griffiths-Jones, S., 157
Grigera, J. R., 166, 291
Grindley, H. M., 52
Grosberg, A. Y., 222
Gross, H. J., 156
Grossman, A., 321
Grotendorst, J., 284
Grubmuller, H., 285
Gruebele, M., 221
Gsponer, J., 226, 228
Gu, L.-Q., 284
Guerra, C., 51
Guex, N., 157
Guillemain, P., 325
Gulbis, J. M., 286
Gulukota, K., 228
Gummel, H. K., 292
Gumport, R. I., 156
Günther, U. L., 325
Guo, H., 164
Guo, J. T., 51
Guo, Q.-X., 329
Guo, W., 225
Guo, Z., 223, 224, 226
Gupta, S. K., 162
Gupta, S. P., 328
Gursoy, A., 165
Gussakovsky, E., 221
Gutin, A. M., 221, 222, 228
Guzowska-Swider, B., 328

Ha, S., 164
Haak, J. R., 166
Haber, E., 220
Haberthur, U., 228
Hackbush, W., 289
Hadley, C., 55
Hageman, J. A., 324
Hai-Lin, S., 324
Halgren, T. A., 291
Hall, C. K., 223, 228
Hall, L. H., 327
Hallick, R. B., 156
Haneef, I., 164
Hänggi, P., 226
Hansch, C., 327, 328
Hansmann, U. H. E., 221
Haran, G., 221
Harata, K., 160
Hardin, C., 156
Hargbo, J., 160
Harms, G. S., 286
Harrington, P. d. B., 322, 327, 328
Harrison, A., 55
Harrison, R. J., 327
Harroun, T. A., 285
Hart, R., 48
Hartl, F. U., 227
Hartley, D. M., 227
Hasegawa, K., 328
Hastie, T., 53
Haussler, D., 159
Havel, T. F., 54, 163
Hawkins, G. D., 225
Hayer-Hartl, M., 227
Hayes, F. R., 164
Hazes, B., 166

Head-Gordon, M., 289
Head-Gordon, T., 166, 224
Heinig, M., 159
Heller, W. T., 285
Hellinga, H. W., 166
Henderson, D., 288, 289, 290
Henikoff, J. G., 161
Henikoff, S., 161
Heringa, J., 162
Heritage, T. W., 328
Herman, M. F., 326
Hermans, J., 290, 291, 292
Hesselbo, B., 221
Heyes, D. M., 289
Higgins, D., 161
Higgins, D. G., 162
Higgins, T., 160
Hilden, H., 52
Hildreth, E., 324
Hill, R. L., 155
Hille, B., 284
Hinton, J. F., 285
Hirs, C. H. W., 161
Hirshberg, M., 225
Hiwatari, Y., 226
Hoang, T. X., 55
Hockney, R. W., 288
Hoffman, K. A., 54
Hofrichter, J., 226
Holladay, N. B., 290
Holland, J. H., 159
Hollerbach, U., 285, 293
Hollich, V., 157
Holm, C., 288
Holm, L., 48, 51, 52, 53, 55
Hon, G., 51
Honeycutt, J. D., 223
Honig, B. H., 53, 166, 293
Hoogland, C., 157
Hörchner, U., 325
Hore, P. J., 221
Horn, B., 52
Horn, H. W., 166
Horn, R., 286
Hou, S., 324
Høy, M., 324
Hoyles, M., 287, 290
Hu, W., 285
Hu, Z., 329
Hu, Z.-Z., 158
Huang, C. C., 161
Huang, H., 157, 158
Huang, H. W., 285
Huang, T. S., 52
Huang, X., 162
Hubbard, B. B., 322
Hubbard, T. J. P., 54, 158
Hubbard, T., 48
Huber, R., 167
Huber, T., 224
Hubner, I. A., 227
Hughey, R., 159
Hukushima, K., 222
Hulo, N., 157
Hummer, G., 226, 287
Humphrey, W., 49, 167, 286
Hünenberger, P. H., 224
Hunt, L. T., 161
Hura, G. L., 166
Hutchinson, E. G., 50, 167
Hutter, H., 322, 323

Ilkowski, B., 223
Im, W., 286, 287, 290
Impey, R. W., 290
Insana, M. F., 325
Islam, S. A., 51, 165
Istrail, S., 51
Ivanyi, I., 157
Iyengar, S. S., 326

Jackson, J. D., 158, 289
Jackson, R. M., 164
Jackson, S. E., 221
Jacoboni, C., 288
Jacobs, A. R., 158
Jacobson, M. P., 166
Jacobsson, S. P., 327
Jakobsson, E., 285, 286, 287, 290, 291, 292, 293
Jalaie, M., 291
Jalali-Heravi, M., 328
Jambeck, P., 156
James, M. N. G., 165
Janin, J., 49, 50, 51, 165
János, P., 326
Jaroszweski, L., 223
Jawerth, B., 325
Jayasuriya, K., 328
Jeanmougin, F., 161, 162
Jennings, A. J., 161
Jennings, P. A., 227
Jensen, L. H., 161
Jernigan, R. L., 49

Jerome, J. W., 292
Jetter, K., 322
Jewett, A. I., 228
Jiang, Y., 286
Ji-Hong, W., 324
Jin, J., 324
Jinping, L., 326
Joannopoulos, J. D., 326
Johal, H., 329
Johnson, A., 286
Johnson, B. R., 326
Johnson, M. S., 158, 163, 164
Jonathon, P., 328
Jones, A., 324
Jones, D. T., 48, 55, 160, 163
Jones, D., 164
Jones, M. L., 50
Jones, S., 48, 55
Jones, T., 159
Jordan, H. F., 289
Jordan, P. C., 285, 286, 288
Jorgensen, W. L., 166, 224, 290, 291
Josefson, M., 324
Joseph-McCarthy, D., 164
Jouan-Rimbaud, D., 323
Jung, J., 53

Kabsch, W., 50, 51, 52, 159
Kaindl, K., 54
Kalé, L., 165
Kalia, R. K., 326
Kalivas, J. H., 325
Kaminsky, G. A., 292
Kaneko, H., 289
Karanicolas, J., 224
Karasawa, N., 289
Karchin, R., 159, 160
Karimi, A., 225
Karplus, K. J., 164
Karplus, K., 159
Karplus, M., 49, 164, 165, 166, 220, 222, 223, 224, 225, 226, 227, 285
Karshikoff, A., 286
Kasif, S., 160
Kateman, G., 322
Kaufman, A., 286
Kautsky, J., 327
Kauzmann, W., 220
Kawabata, T., 53
Kececioglu, J. D., 162
Kell, D. B., 322, 323, 324
Kelley, L. A., 163, 164, 167
Kendrew, J. C., 48, 163
Kestner, N. R., 326
Ketchem, R. R., 285
Khanna, A., 157
Kier, L. B., 327
Kierzek, A. M., 288
Kim, D., 51
Kim, P. S., 220
Kimelman, D., 53
Kimmel, A. R., 156
King, R. D., 163
Kinsey, J. L., 326
Kirkpatrick, S., 167
Kiryutin, B., 158
Klein, M. L., 166, 290
Klevecz, R. R., 324
Kleywegt, G. J., 53
Klimov, D. K., 223, 226, 227, 228
Kloeden, P. E., 292
Klosek, M. M., 226, 292
Koch, M., 322
Koehl, P., 49, 50, 51, 52, 53, 156, 165
Koetzle, T. F., 48
Köküer, M., 329
Kolinski, A., 160, 222, 223
Kollman, P. A., 53, 165, 222, 224
Kolodny, R., 53, 54
Koltun, W. L., 48
Koonin, E. V., 158
Kopp, J., 157
Kosloff, R., 326
Kourtesis, P., 158
Kowalski, B. R., 324
Kraulis, P. J., 48
Krawetz, N., 165
Krienke, H., 287
Krogh, A., 159
Kronland-Martinet, R., 325
Krüger, P., 224
Kryachko, E. S., 326
Krylov, D. M., 158
Kubelka, J., 226
Kubinyi, H., 156
Kuchnir, L., 164
Kuczera, K., 164
Kumar, S., 222
Kunin, V., 55
Kuntz, I. D., 53
Kunz, W., 287
Kuo, A., 286
Kurnikova, M. G., 285, 291, 292

Kusumoto, Y., 227
Kuyucak, S., 285, 287, 290, 291, 293
Kyte, J., 162

Ladenstein, R., 286
Laine, A. F., 325, 327
Lam, P. K., 326
Lamm, G., 293
Lampoudi, S., 225
Lamzin, V., 161
Landau, D. P., 222
Langdon, A. B., 290
Langendijk-Genevaux, P. S., 157
Lantelme, F., 290
Lanteri, S., 322
Larsson, B., 159
Larter, R., 49, 55, 293
Laskowski, R. A., 50, 166, 167
Lasters, I., 166
Lau, F. T. K., 164
Lau, K. F., 222
Laudon, M., 292
Laurents, D. V., 53
Laws, G. M., 284
Lazaridis, T., 225
Le Moigne, J., 325
Leach, A. R., 155, 289
Leahy, R. M., 325
Ledley, R. S., 158
Lee, A., 286
Lee, B., 53
Lee, F., 324
Lee, F. S., 288
Lee, H., 289
Lee, M. S., 224
Lee, W. K., 285
Lehninger, A. L., 156
Leimkuhler, B., 291
Lengauer, T., 156
Leo, A., 327
Leopold, P. E., 221
Lesk, A. M., 49, 50, 54, 157
Letunic, I., 158
Leung, A. K.-M., 322, 325, 326
Levinthal, C., 220
Levitt, D. G., 288, 292
Levitt, M., 49, 50, 51, 53, 54, 156, 157, 165, 167, 225
Levy, R. M., 292
Lewis, J., 286
Li, A., 225
Li, C. H., 329
Li, H., 325
Li, J. P., 329
Li, K.-K., 326
Li, L., 226, 227
Li, Q., 51
Li, S. C., 287
Li, X.-N., 323
Li, Y. X., 49
Liang, Y., 324
Liang, Y.-Z., 323
Liljas, A., 51
Lin, S. L., 52
Lin, X., 324
Lindahl, E., 287
Linden, A., 50
Linial, N., 54
Linn, S. M., 156
Lipkowitz, K. B., 49, 54, 55, 156, 284, 286, 287, 288, 290, 291, 293, 326
Lipman, D. J., 157, 158, 162
Lipman, E. A., 221
Liu, J. K., 50
Liu, L., 329
Liu, M., 329
Liu, W. M., 49
Lo Conte, L., 51
Loan, C. F. V., 51
Lockwood, L., 327
Lomakin, A., 227
Looger, L. L., 166
Lopez, R., 157
López-Ferreras, F., 329
Lorentzen, E., 50
Lorusso, A., 51
Lovell, S. C., 165
Lu, H. P., 286
Lu, N., 324
Lu, X.-Q., 323
Lucier, B. J., 325
Ludwig, C., 325
Lugli, P., 288
Luo, J. C., 51
Luo, R. Y., 50
Luthey-Schulten, Z., 156, 220, 223
Lüthy, R., 53, 165
Lybrand, T. P., 287
Lyubartsev, A. P., 221

Ma, B., 228
MacArthur, D. A., 227
MacArthur, M. W., 166, 167
MacCallum, R. M., 164

MacKerell, Jr., A. D., 164, 165, 291
MacKinnon, R., 286
Madden, T. L., 157
Maddocks, J. H., 54
Madsen, D., 53
Madura, J. D., 166, 290
Maggiora, G. M., 49
Magrane, M., 157
Maigret, B., 165
Mainz, D. T., 292
Mair, G. A., 155
Maldonado-Bascón, S., 329
Mallet, Y., 321, 327
Mamonov, A. B., 291
Mandel-Gutfreund, Y., 159
Manjunath, S., 325
Mannhold, R., 156
Manning, R. S., 54
Mao, J., 323
Maosen, Z., 323
March, C., 164
Marenduzzo, D., 55
Marinari, E., 221
Maritan, A., 55
Mariuzza, R. A., 157
Mark, A. E., 166, 224, 291
Marr, D., 324
Marshall, M., 157
Martens, B., 325
Martens, H., 324
Martin, M. J., 157
Martinov, M., 328
Martins, V. L., 325
Marti-Renom, M. A., 156
Martsinovski, A. A., 221
Martyna, G. J., 291
Marx, D., 284
Mashl, R. J., 293
Mason, J., 329
Massart, D. L., 322, 323, 326, 327
Massi, F., 228
Matkowsky, B. J., 226
Matouschek, A., 227
Matsuo, Y., 53
Matthews, B. W., 53
Mattos, C., 164
Maxwell, A., 156
May, A. C. W., 48, 54
Mayhew, S. G., 160
Mayo, S. L., 166
Mazumder, R., 158
McArdle, P., 160
McCammon, J. A., 285, 292
McCarthy, A., 160
McCarthy, G., 325
McGregor, M. J., 165
McGuffin, L. J., 160
McLachlan, A. D., 52
McLeod, A., 164
McMahon, B., 158
McMillan, N. D., 329
McQuarrie, D. A., 292
Mekhedov, S. L., 158
Meller, J., 160
Melo, F., 156, 157
Meng, E. C., 161
Méra, J., 329
Mermin, N. D., 292
Merz, Jr., K. M., 224, 286
Metropolis, N., 220
Meyer, D. J., 48
Meyer, F. G., 325
Meyer, Y., 321, 324, 325
Mezard, M., 220
Mezey, P. G., 288
Mian, I. S., 159
Micchelli, C. A., 323
Micheletti, C., 55
Michie, A. D., 48, 50, 54, 55
Michnick, S., 164
Michoud, K., 157
Milik, M., 222
Miller, R. T., 160
Miller, W., 157, 162
Milman, J. D., 50
Miloshevsky, G. V., 286, 288
Milpetz, F., 158
Minjun, C., 329
Minkkinen, P., 324
Misiti, M., 322
Misiti, Y., 322
Mitra, S. K., 325
Mitsutake, A., 222
Mittermayr, C. R., 322, 323
Miura, Y., 325
Miyashita, Y., 328
Mizuguchi, K., 54, 157
Mo, J., 323
Mo, J.-Y., 323
Modisette, J. P., 326
Molenaar, J., 292
Möller, S., 159
Molt, K., 322
Monod, J., 48

Montal, M., 221, 286
Montefusco, L., 323, 326
Morales-Cabral, J. H., 286
Morris, A. L., 167
Moss, D. S., 166
Mott, R., 55
Mount, D. W., 156
Mountain, R. D., 290
Moxon, S., 157
Moy, G., 293
Mozayeni, F., 322
Mueser, T. C., 160
Muirhead, G., 49
Muirhead, H., 163
Muller, D. J., 286
Muraki, M., 160
Muramatsu, A., 284
Murenzi, R., 324
Murray, J. S., 328
Murray, W. J., 327
Murtagh, F., 329
Murzin, A. G., 48, 50, 55
Myers, E. W., 157

Nadler, B., 290, 293
Naeh, T., 290
Nagano, N., 50
Nagata, C., 328
Nagle, J. F., 287
Nagy, S., 326
Nakamura, H., 326
Nanba, T., 325
Natale, D. A., 157, 158
Needleman, S. B., 161
Needs, R. M. L. J., 289
Negahdaripour, S., 52
Neher, E., 284
Nelson, D. L., 156
Nemoto, K., 222
Nestorovich, E. M., 286
Neue, G., 325
Neuhaus, T., 221
Ngo, T., 164
Nguyen, D. T., 164
Nguyen, H. D., 228
Nicholas, Jr., H. B., 162
Nicholas, K. B., 162
Nichols, A., 293
Nie, L., 324
Niemöller, A., 322
Nikolaou, M., 329
Nikolov, S. G., 322, 323
Nikolskaya, A. N., 158
Nilsson, L., 165
Nishikawa, K., 52, 53
Nitzan, A., 285, 291
Nochomovitz, Y. D., 223
Nola, A. D., 166
Nonner, W., 288
Noord, O. E. D., 327
Nord, L. I., 327
Nordlander, P., 326
North, A., 49, 155
North, A. C. T., 155
Notredame, C., 162
Novotny, J. A., 291
Novotny, M., 53
Nussinov, R., 52, 228
Nymeyer, H., 221, 222, 227

O'Donovan, C., 157
O'Farrell, P. A., 160
Ochagavia, M. E., 53
Offord, R. E., 50
Ohlson, M., 53
Okamoto, Y., 222, 225
Olafson, B. D., 165, 224
Oliveberg, M., 226, 227
Olson, W. K., 50
Onuchic, J. N., 220, 221, 222, 223, 224, 225, 226, 227
Ooi, T., 52
Oppenheim, G., 322
Orcutt, B. C., 161
Orengo, C. A., 48, 50, 52, 54, 55
Orland, H., 221
Orlandi, E., 222
O'Rourke, B., 329
Orr, G., 286
Orr, T. M., 325
Ouali, M., 163
Ouzounis, C. A., 52, 55
Overington, J. P., 157, 164
Ozdemir, M., 328

Paci, E., 226, 227
Pack, S. E., 328
Page, R. D. M., 156
Palau, J., 50
Pan, Z., 323
Pande, V. S., 222
Parisi, G., 220, 221
Párkányi, C., 328
Parr, R. G., 328

Parry-Smith, D. J., 156
Pastor, R. W., 166
Pauling, L., 48
Pawowski, K., 164
Pearl, F., 55
Pearlman, D. A., 165
Pearson, W. R., 53, 162
Pedersen, L., 289
Peitsch, M. C., 157
Pereira-Leal, J. B., 55
Perera, L., 289
Perham, R. N., 163
Permann, D. N. S., 323
Perozo, E., 286
Perrey, S. W., 162
Perrin, C., 322
Perutz, M. F., 49, 163
Peterson, C., 53
Peterson, K. L., 326
Petsko, G. A., 50, 156
Pettersen, E. F., 161
Pettersson, K., 324
Pettitt, M., 49
Peyrin, F., 326
Pfuetzner, R. A., 286
Phale, P., 287
Phan, I., 157
Philippsen, A., 286
Phillips, D. C., 48, 50, 155
Phillips, J., 165
Phillips, L., 328
Phillips, S. E. V., 157
Piette, B., 324
Pike, A. C., 160
Pilbout, S., 157
Pillardy, J., 160
Pitera, J. W., 166, 225
Platen, E., 292
Plewniak, F., 161
Poggi, J.-M., 322
Pogorelov, T. V., 156
Pogson, C. I., 50
Pohl, P., 285
Pohrille, A., 290
Poirrette, A. R., 52, 53
Politzer, P., 328
Poljak, R. J., 157
Pollock, E. L., 289
Pomes, R., 293
Pommerell, C., 289
Ponder, J. W., 165
Ponting, C. P., 50, 158, 162
Popescu, D. C., 322
Poppi, R. J., 323
Popplewell, A., 225
Postma, J. P. M., 166, 290, 292
Pratt, L. R., 287, 290
Priddle, J. D., 50
Prigogine, I., 227
Prodhom, B., 164
Profeta, S., 224
Prusiner, S. B., 227
Ptitsyn, O., 220
Puccio, L., 323, 326
Pujadas, G., 50
Pullman, B., 290

Qi, J., 329
Qian, X., 291
Quake, S., 321

Raff, M., 286
Rajagopal, G., 289
Ramachandran, G. N., 167
Ramanathan, P. S., 290
Rand, R. P., 285
Randić, M., 327
Ranganathan, S., 328
Rao, B. S., 158
Rao, R. M., 321
Rashin, A. A., 51
Raushel, F. M., 50
Ravaioli, U., 292
Rayment, I., 50
Redaschi, N., 157
Reddy, A. K. N., 287
Reich, S., 291
Reiher, I. W. E., 164
Reimann, C. T., 54
Religa, T. L., 225
Rhem, M., 327
Rhoades, E., 221
Rice, D. W., 52, 53
Rice, S. A., 227, 290, 292
Richards, F. M., 165, 176
Richardson, D. C., 165
Richardson, J. S., 50, 165
Richelle, J., 162
Rick, S. W., 291
Riedel, S., 329
Rigoutsos, I., 52
Riley, R., 227
Ringe, D., 156
Ringler, M., 53

Roache, P. J., 289
Roberts, K., 286
Robinson, A. L., 322
Rodgers, J. R., 48
Rogan, J., 328
Rogen, P., 54
Rogers, P. H., 160
Rokhlin, V., 288
Romanowicz, B., 292
Rose, G. D., 50, 51, 162
Rosenberg, E., 323
Rosenberg, J. M., 222
Rosenblatt, R., 159
Rosenbluth, A. W., 220
Rosenbluth, M. N., 220
Ross, J., 290
Ross, W. S., 165
Rossman, M. G., 49, 51, 53
Rost, B., 55, 162, 163, 164
Rostovtseva, T. K., 286
Rotkiewicz, P., 223
Roux, B., 164, 165, 285, 286, 287, 290, 291, 293
Rowland, J., 323
Rowland, J. J., 324
Rowley, R. L., 289, 290
Roy, S., 162
Rui, L., 324
Russell, R. B., 50, 53, 161
Ruta, V., 286
Rüterjans, H., 325
Rychlewski, L., 164
Ryckaert, J. P., 224
Rzepa, H., 329

Sa, W., 288
Sabata, B., 51
Sakakibara, S., 323
Sakmann, B., 284
Salesin, D. H., 324
Šali, A., 49, 156, 163, 164, 222, 223
Salsbury, F. R. J., 224
Salzberg, S., 160
Sanbonmatsu, K. Y., 225
Sánchez, R., 156, 164
Sander, C., 48, 50, 51, 52, 53, 55, 159, 163
Sansom, M. S. P., 286, 287, 288, 291
Saqi, M. A. S., 161
Saraniti, M., 284
Sarma, V. R., 155
Sasisekharan, V., 167
Sato, S., 225
Saux, V. L., 157
Scawen, M., 225
Schäffer, A. A., 157, 162
Scheraga, H. A., 225
Schimmel, P. R., 48
Schirmer, R. H., 48
Schirmer, T., 286, 287
Schlenkrich, M., 164
Schleyer, P. v. R., 165
Schlichting, I., 161
Schlick, T., 289, 291
Schmidler, S. C., 53
Schmidt, S., 158
Schmitt, E., 326
Schneider, M., 157
Schneider, T., 161
Schnitzer, J., 293
Schonemann, P. H., 52
Schuler, B., 221
Schulten, K., 49, 165, 167, 286
Schultz, J., 158
Schultz, M. H., 289, 290
Schulz, G. E., 48
Schumaker, M. F., 293
Schuss, A., 290
Schuss, Z., 226, 292
Schwartz, R. M., 161
Schwede, T., 157
Scott, H. L., 287
Scott, W. R. P., 224
Searle, S. M., 162
Searls, D., 160
Seeber, R., 327
Seefeld, S., 290
Seibel, G., 165
Sela, M., 220
Selkoe, D. J., 227
Sen, K., 328
Senhadji, L., 325
Seno, F., 55
Seok, C., 52
Serrai, H., 325
Serrano, L., 224, 227
Sfatos, C. D., 228
Shakhnovich, E. I., 221, 222, 223, 226, 227, 228
Sham, S., 285
Shao, L., 52, 325
Shao, X., 322, 323, 324, 325
Shapiro, J. M., 325
Sharfetter, D. L., 292
Sharon, R., 225

Shea, J.-E., 49, 156, 220, 221, 223, 225, 226, 228
Sheeler, D. J., 228
Sheetz, M. P., 50
Sheinerman, F. B., 225
Shelenkov, A. A., 165
Shen, H., 324
Shen, L. D., 288
Shevkunov, S. V., 221
Shew, S. L., 325
Shih, T.-M., 325, 326
Shimada, J., 223, 289
Shindyalov, I. N., 50, 51, 53, 157
Shinozaki, A., 165
Shu, C., 292
Sibanda, B. L., 158
Siddiqui, A. S., 51
Sieker, L. C., 161
Sielecki, A. R., 165
Sierk, M. L., 53
Sigrist, C. J. A., 157
Simonin, J.-P., 287
Singh, U. C., 224
Sippl, M. J., 54, 160, 165, 167
Sjölander, K., 159
Sjöström, M., 322
Skeel, R. D., 291
Skell, R., 165
Skolnick, J., 160, 222, 223
Smirnov, S., 158
Smit, B., 166
Smit, H. C., 322
Smith, A. V., 223
Smith, G. R., 286, 288
Smith, J. C., 164
Smondyrev, A. M., 287
Snaith, P. J., 328
Snow, M. E., 163
Snowden, C. M., 292
Socci, N. D., 220, 222, 223, 226, 227
Sokolowski, S., 288
Sonnhammer, E. L. L., 157, 159
Sorenson, J. M., 224
Sowdhamini, R., 158
Spassov, V., 286
Spellmeyer, D. C., 224
Srinivasan N., 158, 163
Srinivasan, S., 164
Stanley, H. E., 223, 226
Stasiak, A., 54
States, D. J., 165, 224
Stebbings, L. A., 157
Stein, L. D., 156
Steipe, B., 54
Stern, H. A., 292
Sternberg, M. J. E., 49, 51, 158, 161, 162, 164, 165
Steward, A., 226
Stewart, M., 55
Stinchcombe, R. B., 221
Stollnitz, E. J., 324
Stork, C. L., 324
Stormo, G. D., 156
Stote, R., 164
Stouch, T., 288
Stout, G. H., 161
Stoye, J., 161, 162
Straatsma, T. P., 166, 291
Strandberg, B., 48
Straub, J., 164
Straub, J. E., 222, 225, 228
Streppel, M., 324
Stroud, R. M., 50
Stryer, L., 287
Stuart, A. C., 156
Stuart, S. J., 291
Stubbings, T., 323
Studholme, D. J., 157
Su, Q., 323
Subbiah, S., 53
Subramaniam, S., 285, 287, 292
Sudarsanam, S., 164
Sugar, I. P., 287
Sugita, Y., 222, 225
Sukumar, N., 327, 328
Sun, P., 323, 324
Sundling, C. M., 327
Sutcliffe, M. J., 163, 164, 167
Sutmann, G., 284
Sutton, B., 225
Suyama, M., 53
Suzek, B. E., 158, 285
Sverdlov, A. V., 158
Swaminathan, D. J., 224
Swaminathan, S., 165
Sweet, R. M., 158, 161
Swendsen, R. H., 222
Swindells, M. B., 48, 51, 54, 55
Swope, W. C., 166, 225
Syganow, A., 290
Szabo, A., 166
Sze, S. M., 284
Szustakowski, J. D., 53

Takada, S., 223
Takahashi, K., 52
Taketomi, H., 222
Talkner, P., 226
Tan, H.-W., 323
Tanaka, T., 222, 289
Tang, J., 284, 292
Tang, Y., 53
Tang, Y. Y., 293, 329
Tang, Z., 323
Tapia, O., 54
Tatera, J. F., 322
Tatusov, R. L., 158
Taylor, W. R., 48, 49, 51, 52, 160, 162, 163
Teichmann, S., 55
Teitelbaum, H., 323
Teller, A. N., 220
Teller, E., 220
Teodorescu, O., 160
Teolis, A., 321
Teplow, D. B., 227
Teppola, P., 324
Tesi, M. C., 222
Teter, M. P., 326
Teubner, B. G., 289
Thirumalai, D., 222, 223, 226, 227, 228
Thirup, S., 159
Thomas, P. D., 220
Thompson, J., 161
Thompson, J. D., 162
Thompson, T. E., 287
Thornton, J. M., 48, 50, 54, 55, 158, 160, 163, 164, 166, 167
Thornton, T., 284
Thrall, B. D., 286
Tieleman, D. P., 285, 286, 287, 288
Tildesley, D. J., 166, 221, 288
Timasheff, S. N., 161
Timberlake, K. C., 49
Timmerman, H., 156
Tirado-Rives, J., 224
Tironi, I. G., 224
Tisdall, J. D., 156
Tobi, D., 160
Tönges, U., 162
Tooze, J., 48, 155
Topham, C. M., 164
Torda, A. E., 224
Torrésani, B., 322
Torrie, G. M., 227, 291
Toukmaji, A. Y., 289
Townsley, L. E., 285
Trovato, A., 55
Trsitram-Nagle, S., 287
Truhlar, D. G., 225, 328
Trygg, J., 322, 328
Tsai, J., 225
Tsallis, C., 221
Tu, S., 159
Tucker, A., 285
Tucker, W. A., 285
Tuckerman, M., 291
Tuparev, G., 52
Turner, D. B., 328
Turq, P., 287, 290
Tycko, R., 227
Tymoczko, J. L., 287

Ueda, Y., 222
Ullman, A. H., 322
Ulrici, A., 327
Umeyama, S., 52
Unser, M. A., 325
Urbas, A., 327
Urry, D. W., 285

Valleau, J. P., 227, 291
van den Bogaert, B., 327
van den Heuvel, E. J., 322
van der Straaten, T., 292
van Gunsteren, W. F., 166, 224, 290, 292
van Rensberg, E. J. J., 222
Vanaman, T. C., 155
Vandeginste, B. G. M., 322
Vandewalle, S., 289
Varadarajan, K., 165
Varma, S., 286, 292
Vashishta, P., 326
Vasudevan, S., 158
Vecchi, M. P., 167
Vega, M. C., 50, 224
Vel, O. D., 327
Veltkamp, D. J., 322, 324
Vendruscolo, M., 226, 227
Veretnik, S., 50
Verlet, L., 224, 290
Vinayaka, C. R., 158
Virasoro, M. A., 220
Vojtechovsky, J., 161
Voltz, R. A., 52
von Heijne, G., 159
von Kitzing, E., 290, 326
Voronoi, G. F., 167
Vorontsov-Velyaminov, P. N., 221

Vrielink, J. A. M., 327
Vriend, G., 52
Vuthandam, P., 329

Walczak, B., 322, 323, 326, 327
Waley, S. G., 50
Walker, M., 290
Walker, M. W., 52
Walker, N. S., 51
Wallace, A. C., 50
Wallace, B. A., 285
Wallqvist, A., 290
Walsh, D. M., 227
Walsh, M. A., 160, 161
Walter, P., 286
Wan, C., 327
Wan, T., 225
Wang, F., 222, 324
Wang, G., 50, 161
Wang, J., 324, 326
Wang, J. S., 222
Wang, L., 224
Wang, X. Z., 324
Warshel, A., 288
Watanabe, M., 164, 291
Watenpaugh, K. D., 158
Waterfield, M. D., 54
Watson, H. C., 163
Weaver, D. L., 220
Wehrens, R., 324
Wei, L., 326
Wei, S., 326
Weiner, P., 224
Weiner, S. J., 224
Weiss, T. F., 293
Weiss, T. M., 285
Weissig, H., 48, 157
Welsh, W. J., 329
Wen-Can, C., 324
Weng, Z. P., 53
Westbrook, J., 48, 157, 158, 159
Westhof, E., 156
Wetlaufer, D. B., 220
White, C. A., 289
White, J. H., 54
White, Jr., F. H., 220
Whittington, S. G., 222
Wickerhauser, M. V., 321, 322, 325, 329
Wiederstein, M., 167
Wierenga, R. K., 50
Wilk, S., 284
Will, G., 49
Willett, P., 52, 53, 328
William, G., 48
Wilmanns, M., 50, 224
Wilmot, C. M.
Wilson, I. A., 50
Wilson, K. S., 161
Wilson, M. A., 290
Winson, M. K., 323
Wiórkiewicz-Kuczera, J., 164
Wipke, W. T., 323
Wise, B. M., 324
Wise, E. L., 50
Wise, J. A., 156
Wodak, S. J., 51, 53, 162, 165
Wold, S., 322, 328
Wolf, T. B., 287
Wolf, Y. I., 158
Wolfson, H. J., 52
Wolkenstein, M., 323
Wolynes, P. G., 220, 223, 226, 228
Won, Y., 165
Woodward, A. M., 322, 323
Word, J. M., 165
Wordeman, M., 292
Workman, J. J., 322
Wright, P. E., 225
Wu, C. H., 157, 158
Wu, S., 324
Wu, T. D., 53
Wunsch, C. D., 161
Wunsch, P., 327

Xiang, Z., 166
Xu, D., 51
Xu, Y., 51
Xue, H., 329

Yanai, T., 327
Yang, A. S., 53
Yang, L., 285
Yang, W., 328
Yang, X. W., 329
Yang, Y., 290
Yeates, T. O., 167
Yeats, C., 157
Yee, D. P., 220
Yeh, L.-S. L., 157, 158
Yellen, G., 286
Yin, D., 164
Yin, J. J., 158
Yiyu, C., 329
Yi-Zeng, L., 324

Yonezawa, T., 328
York, D., 289
Young, A. P., 220
Young, D. M., 289
Yue, K., 220
Yuen, P. C., 329

Zaim, M., 325
Zavorin, I., 325
Zehfus, M. H., 51
Zhang, C. T., 49
Zhang, J., 157, 158
Zhang, M., 324
Zhang, R., 329
Zhang, X., 324, 329
Zhang, Z., 157, 325
Zhao, G., 324, 325
Zhao, J., 329
Zheng, J., 324
Zheng, L., 24
Zhou, G. P., 50
Zhou, R., 226, 292
Zhou, T., 53
Zhou, Y., 223, 286
Zielinski, W. L., 329
Zvelebil, M. J., 54, 162

Subject Index

Computer programs are denoted in boldface; databases and journals are in italics.

Ab initio protein folding, 61
Ab initio quantum chemistry, 315
AB model, 184
Accessible surface area, 15
Activity coefficients, 240
Adjacent gaps, 98
Adjacent side chains, 118
Admissibility condition, 301
Algorithms for domain identification, 39
Aligned substructures, 29
Alignment, vii, 58, 90
Alignment order, 101
Alignment scores, 90
ALSCRIPT, 112
Alzheimer's disease, 219
AMBER, 124, 133, 192, 200
Ambiguous electron density maps, 141
Amino acid sequences, 60, 116, 232
Amino acids, 5
Amphipathic molecules, 236
Amplitude information, 297
α-Amylase, 11
Analytical models, 181
Annotation-based searches, 61
Applications of wavelets in chemistry, 309
Aqueous solutions, 231, 239
Aristotle, 2
Artificial charge distribution, 247
Artificial neural networks (ANN), 70, 73, 313
ASTRAL, 13, 39
Atomic Non-Linear Environment Assessment (ANOLEA), 63
Atomic volumes, 144
Atomistic models, 241
Atomistic simulation, vii
Authors, 16
Autocorrelation function, 180
Automated alignment methods, 101
Automated classification, 40
Automatic identification of protein domains, 14
Average concentration, 281
Average crossing number, 34, 35
Average electric fields, 281
Average group clustering, 47
Average linkage clustering, 47
Average linkage hierarchical clustering, 43

Back-bone chain, 35
Backbone-dependent rotamer library, 126
Background signal, 309
Backtracking trees, 130
Bacteriorhodopsin, 8
Barrel structure, 10
β-Barrels, 10, 187, 235
Basis pdfs, 114, 115
Basis set expansions, 315
Basis set superposition error (BSSE), 315
Berendsen external heat-bath method, 134
Best basis, 307, 313
Best conformation, 129
Best model, vii, 60, 84, 124, 137
Bias, 116, 170
Biased sampling, 198
Biasing potential, 197, 216
Binding sites, 86, 89, 121, 174

Reviews in Computational Chemistry, Volume 22
edited by Kenny B. Lipkowitz, Thomas R. Cundari, and Valerie J. Gillet

Binomial system of nomenclature, 2
Bioactive sequence, 89
Biogeometry, 3
Bioinformatics, 3, 60, 61
Biological channels, ix
Biological ion channels, 229
Biological macromolecules, 1
Biological molecules, v, vi
Biological time scales, 174, 187, 265
Biology, xi
Biotech Validation Suite, 13
Blast, vii, 63, 120
Block substitution matrix (BLOSUM) matrices, 91, 95, 140
BLOCKS, 39
Boltzmann probability, 185
Boolean search, 70
Born-Oppenheimer approximation, 191
Boundary conditions, x, 231, 246, 250, 261
Bovine α-lactalbumin, 58, 73, 83, 85, 122
Branch-and-bound backtracking, 129
Brownian dynamics, x, 236, 239, 247, 264, 267
Bulk macroscopic properties, 268
Bulk properties, 243
Bulk water, 239
Buried protein atoms, 145
Buried regions, 108

Cα-Cα distance maps, 14
California Quail Lysozyme C, 64
Calorimetrically determined enthalpies, 176
Cambridge Structural Database (CSD), 138
Canonical ensemble, 200
Capillary electrophoresis, 311
Carbohydrates, 1, 68
CASP project, 33, 72
CE, 25
Cell membrane, 236
Cell membrane channels, 9
Cells, 2
Cellular functions, 3
Chain connectivity, 174
Chain threading, 35
Channel proteins, 232
Chaperone-mediated folding, 219
Chapman-Kolmogorov equation, 275
Charge cloud, 251
Charge conservation, 278
Charge continuity equation, 278
Charge distribution, 276
Charge shape, 250
CHARMM, 115, 124, 133, 192, 198
Chebyshev norm, 32
Chemical information, 296
Chemical potential, 262
Chemical spectroscopy, 296
Cheminformatics, xi, 295, 296, 316
Chemometrics, xi, 296, 316
Chicken lysozyme, 83, 122, 139, 151
Chirality, 117, 138
Cholesterol, 237, 238
Chromatography, 311, 313
Class, 38
Class, Architecture, Topologies, and Homologous (CATH) classification, vi, 3, 16, 32, 39, 42, 44
Classification, vi, xi, 2, 231, 316
Classification in biology, 3
Closely related sequences, 98
Closure events, 233
Cloud-in-cell (CIC) charge, 251
Clustal, vii, 95, 122
ClustalW, 95, 99, 102
ClustalX, 95, 97, 102, 107
Cluster analysis, 14
Clustering, 32, 47, 208, 316
Clusters of Orthologous Groups (COG), 39, 67
CluSTr, 39
Coarse-grained protein models, 181
Coiflet wavelet, 305
Coiled coil regions, 107
Collagen, 7
Collapsed state, 186
Combinatorial conformer problem, 127
Commitment probability, 204, 206, 208, 211
Common ancestry, 40, 63
Common evolutionary origin, 41
Compaction time, 186
Comparative protein modeling, vii, 57
Complete linkage clustering, 47
Computational artifacts, 261
Computational biology, 61
Computational chemistry, x, 61, 296
Computer particles, 263
Computer simulations, 171
Concentration gradient, 274, 280
Conditional probability, 210
Conditionally convergent series, 247
Configurational entropy, 174
Conformational analysis, 125
Conformational changes, 233, 235
Conformational clustering, 205

Conformational space, 170
Conformational states, ix
Conformations, 170, 179
Consensus fold, 44
Consensus model, vii, 121
Conserved domain, 66
Conserved Domain Architecture Retrieval Tool (CDART), 67
Conserved domain database (CDD), 67
Conserved regions, 121
Constructing protein models, 111
Contact maps, 29
Continuity equation, 274, 277
Continuous Fourier transform (CFT), 297
Continuous wavelet transform (CWT), 301
Continuum model, 274, 281, 283
Continuum of wavelet dilations, 303
Convection, 277
Cooperative folding process, 171, 190
Coordinate root mean square deviation (cRMS), 17, 27, 31
Core regions, 15
Core structures, 120, 146
Corey-Pauling-Koltun (CPK) models, 4
Correct protein structures, 141
Correctly folded protein, 138, 149
Correctly threaded matches, 82
Correlation matrix, 18, 19
Correspondence length, 24
Correspondence, 24
Coulomb force, 244, 258
Creutzfeld-Jacob disease, 219
Critical nucleus, 208, 210, 217
Curated classification, 40
Current conservation, 278
Current density, x, 274, 276, 278
Current density vector, 274
Curse of dimensionality, 316
Cytochrome C, 195

DALI Domain dictionary (DDD), vi, 16, 43
DALI Fold Classification, 39
DALILIGHT, 25
Darwin, 2
Data alignment, 317
Data archiving, 313
Data mining, 14
Data variance, 316
Databases, 1, 61
Databases of protein structural domains, 16
Data-driven discovery, 3
Daubechies 6 wavelet, 300
DDBASE, 16
Dead-end elimination (DEE), 127
Debye length, 281
Decoy structures, 79
3Dee, 16
DeepView, 64
Defective protein, 145
Definition of Secondary Structure of Proteins (DSSP), 71
DEJAVU, 25
Denaturant, 178
Denatured protein, 170
Denatured state, 204
Denaturing conditions, 209
Denaturing simulations, 197
Denoising, xi
Denoising algorithm, 310
Density of states, 180, 182, 197
Descriptors, 33, 316
Descriptors of chemical data, 320
Detective, 16, 42
Deviations in atomic volumes, 145
DIAL, 16
Dielectric barrier, 243
Dielectric constant, 192, 259, 263, 268, 278
Dielectric discontinuities, 250
Dielectric medium, 239
Diffusion coefficient, 239, 263, 268, 274
Diffusion collision model, 171
Diffusivity, 234
Dilated wavelet, 301, 302
Dilation variable, 303
Dilations, 302
Dimension reduction, 317
Dirichlet boundary condition, 262
Discontinuous molecular dynamics, ix
Discrete Fourier transform (DFT), 298, 299
Discrete wavelet transformation (DWT), 303
Discretization errors, 280
Discretization grid, 249
Disjointed signal, 306
Dissimilarity, 47
Distance ALIgnment (DALI) algorithm, 4, 28
Distance geometry, 28
Distance map, 28
Distance matrices, 24
Distance root mean-squared deviation (dRMS), 27, 31
Distance-dependent dielectric, 192
Distance-geometry, viii, 113
Distantly similar sequences, 84
Distantly related proteins, 60

Distinguishable state, 182
Disulfide bonding, 89, 114, 127, 138
Disulfide bridges, 58, 126
Divide-and-Conquer, vii, 100
DNA, 35
Domain assignments, vi, 43
Domain classification, 68
Domain quality, 15
Domain sequence, 81
Domain-based pairwise alignment, 147
DomainParser, 16
Domains, 12, 38, 62, 146
DOMAK, 16, 42
Double-zeta basis sets, 314
DSSP, 13
Dyanmic programming, 24, 95
Dynamic variable, 200

EBGHSTL, 71
Effective properties, 272
Electrical forces, 252
Electrodiffusion of ions, 274, 278
Electrodiffusive continuum, 270
Electron density gradients, 318
Electron density Laplacian, 318
Electron density maps, 116
Electron device simulation, 243
Electronegativity equalization scheme, 273
Electronic kinetic energy densities, 318
Electrophysiologic experiments, 231
Electrophysiology, 230
Electrostatic boundary conditions, 262, 271
Electrostatic moments, 272
Electrostatic potential, 250, 318
Electrostatic potential energy, 247
Electrostatics, 243
eMOTIF, 39
Empirical energy functions, 79
ENCAD, 195
Energy conservation, 278
Energy landscape theory, ix, 170, 172
Energy minimization, 132, 240
Engineering, x
Ensemble of single-chain conformations, 175
Ensemble of target proteins, 113
Ensembles of rotamers, 126
Entrez, 13
Entropy, 174
Entropy crisis, 174
ENZYME, vii, 62
Enzyme Committee (EC) number, 63
Equations of motion, 188, 273
Equilibrium molecular dynamics, 268
Equilibrium conditions, 195
Equilibrium distribution functions, 210
Equilibrium fluctuations, 209
Equilibrium sampling, 180
Ergodicity, 181
ERRAT, 138, 141, 147
Error, 280, 317
Error reduction, 254
Euclidian distances, 27
Euler integration, 265
Evaluating protein models, 138, 148
E-values, 66, 67
Evolutionary conserved residues, 90
Evolutionary distance, 80, 84, 90, 92
Evolutionary distant proteins, 70
Evolutionary history, 84
Evolutionary origin, 41
Evolutionary relatedness, 40
Evolutionary relationships between proteins, 35, 38, 57, 96
Evolutionary trends, 86
Ewald summation methods, x, 244, 247
Exact partition function, 181
Excess chemical potential, 262
Excluded volume, 182, 186
Expected value, 68
Expert Protein Analysis System (ExPASY), 62
Explicit solvation molecules, 137, 200, 267
Exposed regions, 108, 119
Extending gaps, 140
External boundary conditions, 263
External stimulus, 232

Factor analysis, 18
False positive, 65
False relationships, 317
Families of Structurally Similar Proteins (FSSP), 43
Family, vi, 38
Fast archiving, 313
Fast Fourier Transform (FFT), 249, 297
Fast multipole method (FMM), x, 244
FASTA, 99
FATCAT, 25
Feature isolation, 306
Feature pdf, 114, 115
Feature reduction, 317
Feature vectors, 35
Fibrous proteins, 7
Fick's law, 274

Finite differences, 278
Finite difference grid, 250, 280
Finite difference iterative schemes, 252
Finite state machine, 73
Flawed models, 138
Flickers, 233
Fluctuating charge (FQ) model, 272
Fluctuation dissipation theorem, 188
Fluctuations, 277
Flux of charges, 274
Flux-based simulation, 239, 273
Fokker-Planck equation, 275
Fold, vi, 38, 40, 41
Fold families, 35, 42
Fold overlap problem, 44
Fold recognition, 33
Folded protein, ix
Folded state, 175
Folding class, 10
Folding free energy barrier, 216
Folding kinetics, 12, 175, 183, 192
Folding nucleus, 207
Folding pathway, 197, 209
Folding process, ix
Folding progress, 179
Folding rate, ix, 170, 178
Folding routes, 175
Folding temperature, 175
Folding thermodynamics, 175, 183
Folding times, 174, 181, 186
Folding trajectories, 207, 208
FoldMiner, 25
Force, 250
Force fields, 125, 192, 268, 271
Force field parameterization, 192
Force field parameters, 271, 272
Four-helix bundles, 10
Fourier filtering, 309, 311
Fourier transform (FT), xi, 247, 296, 297
FRAGFINDER, 71
Fragment matching, 24
Fragment-base alignment, 87
Framework model, 171
Free energy, 176, 179, 270
Free energy calculation, 270
Free energy minimum, 216
Free energy surfaces, ix
Frequency domain, 297
Frequency information, 297
Friction coefficient, 264, 265, 281
Frustration, 173, 175, 214
Fukui function, 318
Full multigrid method, 257
Fully atomistic simulations, 190
Functional diversity, 11
Functional genomics projects, 3
Funnel-shaped free energy landscape, 170, 174
ϕ-value analysis, ix, 201, 212

Gabor transform, 298
Gap, 23, 27, 68, 76, 87, 105, 110, 118
Gap penalty, 30, 90, 95
Gap residue parameters, 79
Gap-extending penalty (GEP), 98
Gapless fragments, 72
Gap-opening penalty (GOP), 98
Gapped alignment, 67
Gating, 242, 281
Gating ring, 235
Gauβ-Seidel Method, 253
Gaussian multiwavelet basis, 315
GB/SASA, 192, 194
GenBank, 65
Gene sequence, 89
GeneDoc, 107, 122, 123
Generalized Born (GB) model, 192
Generalized ensemble methods, 180
Genetic algorithm, 24, 72, 319
Genetic algorithm/Partial least squares (GA/PLS), 319
Genetic code, 9
Genetic information, 3
Genome, v, 3, 61
GenTHREADER, 110
Geometric hashing, 24
Geometric properties, 33
Geometric similarity, 17
Glass transition temperature, 173, 175, 190
Global alignment, 66, 79, 86, 91, 99
Global energy minimum, 174
Global fluctuations, 209
Global minimum energy conformation, 125
Globin fold, 10
Globular proteins, 7, 9
Glycolipids, 237, 238
Gō-models, 190, 208
Gō-type potentials, 217
Gonnet matrices, 91, 95, 102, 140
Gramicidin A, x, 232
Grand canonical ensemble, 262
Graph theory, 132
Greediness, 88, 99
Greek key barrels, 10

Green's function, 245
Grid, 279
GROMOS, 124, 133, 192
GROMOS96, 63
Guide tree, 95

Haar wavelet, 305
Hartree-Fock equations, 315
Hartree-Fock exchange, 315
Helical proteins, 198
Heme group, 10
Hemerythrin, 10
Hemoglobin, 9, 35
Hen's egg-white lysozyme, 58, 85
Heteropolymer, 5
Heteroscedastic noise, 309
Heuristic approaches, 30
Heuristic search, 65
Hidden Markov models (HMMs), vii, 70, 73
Hierarchic classification, 40
Hierarchical clustering, 47, 197
Hierarchical simulation strategy, 283
High performance computing, 230
High-frequency noise, 309
Hinged proteins, 125
HIV reverse-transcriptase, 319
Homologous protein structures, 118
Homologous proteins, 113
Homologous sequences, 71
Homology modeling, vii, 57
Homoscedastic noise, 309
HOMSTRAD, 39, 62
Horse hemoglobin α, 58
Horse hemoglobin β, 58, 151
HP models, 182
HT model, 189
Human α-lactalbumin, 151
Human genome project, v
Human proteome, v
Hydrated ion, 235
Hydrated membrane/channel system, 242
Hydration shell, 239
Hydrogen bonds, 8, 28, 71, 138, 199, 233
Hydropathy index, 105
Hydropathy plots, vii, 108
Hydropathy profile, 108
Hydropathy score, 108
Hydrophilic amino acid residues, 98
Hydrophilic region, 98, 108
Hydrophobic collapse model, 171
Hydrophobic core, 7, 8, 10, 183, 198, 200
Hydrophobic effects, 174
Hydrophobic interactions, 190
Hydrophobic regions, 108
Hydrophobic residues, 108, 171, 183, 188
Hydrophobic sheets, 198
Hydrophobic sidechains, 232
Hydrophobic thickness, 238
Hypothesis-driven research, 3

Image charges, 282
Immunoglobins, 11
Implicit membrane models, 240
Implicit solvent treatment, 133, 192, 264
Implicit water models, 240
Importance sampling, ix, 197
Improving alignments, 104
In vivo folding, 219
Incorrect stereochemistry, 143
Incorrectly folded proteins, 138, 149, 219
Incorrectly threaded matches, 82
Informatics, x
Information, 301
Information compression, 295
Information cost, 308
Information entropy, 313
Information-rich descriptors, 317
Infrared spectral analysis, 320
Infrared spectral libraries, 313
Infrared spectroscopy, 311, 313
Inhomogeneous charge distributions, 262
Integration time step, 265
Internal distances matrix, 28
Internal electrostatic interactions, 263
International Union of Biochemistry and Molecular Biology (IUBMB), 63
InterPro, 39
Inverse wavelet transform, 302
Ion channel simulation, 230
Ion channels, ix, 229, 231
Ion permeation, 268
Ion pump, 9
Ion transport, ix, 8
Ionic charge transport, 229
Ionic concentration, 274
Ionic drift, 274
Ionic flux, 231, 276
Ionic permeation, 281
Ionic velocity, 277
Irregular property distributions, xi, 295
Ising model, 15
Isoelectric point, 108
Iterative methods, 254
Iterative Search, 69

JalView, 107, 112
3D-JIGSAW, 113, 119
Jelly roll barrels, 10
J-walking, 180

K2, 25
K2SA, 25
K-channel, 242
KcsA channel, 234
Kendrew models, 4
Keratin, 7
Kinetic ϕ-values, 213
Kinetic traps, 183
K-means clustering, 47
Knot theory, 33, 35
Knowledge base, 113
Knowledge discovery, 14
Knowledge-based evaluation, 150
Knowledge-based potentials, 78
Knowledge-based rules, 121
Kohlrausch's law, 274
KvAP channel, 234

Lactose intolerance, 85
Lagrange multipliers, 18
Langevan dynamics, ix, 211
Langevin equation, 188, 264, 265, 275
Langevin temperature equilibration, 134
Large proteins, 12
Large-scale fluctuations, 189
Latent frustration, 175
Lattice models, ix, 171, 179, 182
Lattice Monte Carlo simulations, 181
Lattice move sets, 185
Lattice site, 182
Learning, Observing and Outputting Protein Patterns (LOOPP), 78
Lennard-Jones potential, 188, 190, 192, 259
Levinthal paradox, ix, 170, 174
Like contacts, 184
Linnaeus, 2
Lipid bilayer, 231, 234, 236, 238
Lipid membrane, 229
Lipid mobility, 237
Lipid molecules, x
Lipid/protein interface, 241
LOAD (Library of Ancient Domains), 67
Local alignment, 79, 86, 99
Local average ionization potential, 318
Local energy minima, 180
Local fluctuations, 209
Local free energy minimum, 175
Local geometry matching, 24
Local interactions, 172
Local polarization fields, 283
Local resolution, 314
Local secondary structure, 28, 78
Local similarity, 24, 67, 84
Localized electric fields, 244
LOCK2, 25
Long-range electrostatics, 231
Long-range force, 244
Long-range interactions, x, 247
Long-time process, 194
Loops, 90, 105, 235
Loop regions, 11, 58, 78, 111, 120
Loop segments, 119
Low energy sequence, 183
Low-energy collapsed state, 186
Low-energy conformations, 135
LSQRMS, 25

Machine learning, 313, 316
Macomolecular Crystallographic Information File (mmCIF), 69
Macroscopic polarization behavior, 268
Main-chain conformation, 117
Mainly β proteins, 9, 40
Many-body effects, 272
Markov models, 70, 71
Markov transition model, 28
Markovian random forces, 264
Mass spectrometry, 311, 313
Matching segments, 117
MATRAS, 25
Maximal common subgraph detection, 24, 65
Maximal segment pair (MSP), 67
Mean field approximation, x, 24, 274, 281
Mean structural properties, 274
Measures of similarity, 26, 27
Mechanical wire model, 58
Mechanical work, 235
Mechano-sensitive channels, 232
Melting curves, 176
Membrane, 231, 236
Membrane potential, 234
Membrane proteins, 7
Membrane-spanning pore, 235
MEMSAT, 110
Metafolds, 44
Metrics, 17
Metropolis Monte Carlo method, 171, 186
Meyer wavelet, 305

Mirror transformation, 29
Misalignment, 99, 101
Misaligned regions, 63
Misfolded compact states, 183
Misfolded proteins, 78
Misfolded regions, 63
Misplacement of side chains, 141
Mixed α-β proteins, 9, 40
Mobile ions, 243
MODELLER, 113, 119, 121, 122, 123, 187
MOE, 113, 119, 121, 122
Molecular dynamics, ix, 125, 133, 135, 147, 181, 199, 235, 236, 247, 267
Molecular mechanics, 115, 125, 132
Molecular pdf, 114, 115
Molecular superposition methods, 311
Molecules to Go, 13
MOLSCRIPT, 4
MONSSTER, 79
Monte Carlo, ix, 24, 181, 186, 268
Monte Carlo step, 186
Most homologous template, 110
Mother wavelet, 301, 311
Motif-based secondary structure prediction, 110
Move set, 183, 185
MSD, 13
MthK channel, 234
Multicanonical sample, 180, 212
Multidomain protein structures, 14, 42
Multigrid iteration, 257
Multigrid methods, x, 254, 256, 280
Multiple folding nuclei, 209
Multiple folding pathways, 172
Multiple sequence alignment, 90, 100, 110
Multiple sequences, 84
Multiple template methods, vii, 65, 70, 72, 113, 118
Multipole expansion, 245
Multiresolution analysis (MRA), 304, 309, 312
Multistate folders, 177
Multistate models, 176
Mutants, 230, 236
Mutant channels, 243
Mutated residues, 89
Mutation, 90, 91, 212
Mutation probability scores, 91
Myoglobin, 4, 9, 35, 86

NAMD, 124, 133
Narrow channels, 282
National Center for Biotechnology Information (NCBI), 65
Native conformation, 143
Native contacts, 190, 207, 214
Native state, vii, viii, 182, 189, 204
Native structure, 7
NCBI-BLAST, 65
Nearest-grid-point (NGP) charge, 251
Neighbor-Joining (NJ) tree, 87
Nernst-Planck equation, x, 274, 278
Nest iteration method, 257
Neumann method, 262
Newtonian dynamics, x
Newtonian mechanics, 265
Newton's equations of motion, 172, 191
NMR-based protein structures, 69
NMRCLUST, 138, 146, 147
NMRCORE, 146, 147
Noise, 23, 296, 309
Noise of changing variance, 309
Noise types, 310
Noise wavelets, 309
Noncooperative folding mechanism, 189
Nonlinear wave functions, 314
Non-native conformations, 183
Non-native state minima, 180
Nonoptimal stereochemistry, 138
Nonperiodic boundary conditions, 262
Nonperiodic functions, 298
Nonpolar amino acid side chains, 8
Non-Redundant (NR) database, 65
Nonstationary signals, 298
NRL_3D, 13
Nuclear magnetic resonance (NMR) spectroscopy, 9, 23, 68, 232, 313
Nucleation, 207
Nucleation condensation model, 171, 194
Nucleic acids, 1, 68
Nucleic Acid Research, 61
Number of native contacts, 205

Off-lattice models, ix, 171, 172, 179, 187
OLDERADO, vii, 138, 146, 147, 148
Opening gaps, 140
OPLS, 192
Optimal alignment, 14, 23, 101
Optimal correspondence, 24
Optimal fit bias, 31
Optimal fitting, 301
Optimal signal representation, 308
Optimal wavelet, 306, 311
Optimally aligned residues, 86

Optimization method, 135
Order parameters, 203, 214
Organelles, 2
Ornstein-Zernike equation, 262
Orthogonal wavelet, 305
Outliers, 27, 31

Pairwise alignment, 90
Pairwise residue matches, 99
Pairwise superposition, 146
PAM1, 92
PAM250, 92
Parallel tempering, 180
Parsimonious models, 317
Partial least squares (PLS), 319
Particle-based simulations, 263
Particle-mesh Ewald (PME) method, 249
Particle-Particle-Particle-Mesh (P3M) method, 244
Partition function, 182
Patch-clamp fluorescence microscopy, 233
Pattern recognition, 316, 320
Pattern-Hit Initiated BLAST (PHI-BLAST), 66
PDB at a Glance, 13
PDB ID, 69
PDB90, 43
PDBSum, 13
PDP, 16
Penalty functions, 147
Peptide bond, 5
Peptides, 68
Periodic boundary conditions, 243, 246, 261
Periodic systems, 249
Pfam, 39, 62
Phase space, 263
Phase transition, 173, 174
PHD, 110
pH-gated channels, 236
Phosphatidylcholine, 237
Phosphoglycerides, 237
Phospholipids, 237
Phospholipid bilayer, 7
Photoacoustic spectroscopy, 311
PHYLIP, 96, 97
Phylogenetic trees, 96, 99
phylogeny, 2
Physicochemically similar proteins, 84
PISCES, 13
Pittsburgh Supercomputer Center, 122
Plasma simulations, 263
Point dipole (PD) model, 272
Point mutation, 212
Point-Accepted Mutation (PAM) matrices, 91, 92, 140
Poisson-Boltzmann equations, 192
Poisson-Nernst-Planck (PNP) method, x, 278
Poisson's equations, x, 231, 245, 248, 252, 275
Polarizable-SPC (PSPC) model, 272
Polarization, 272, 281
Polarization field, 272
Polypeptide approximate conformation, 117
Polypeptide loops, 122
Polysaccharides, 85
Pores, 232
Porin channels, 239
Porins, x, 8, 235
Position-specific scoring matrix (PSSM), 66
Postsmoothing, 256
Potassium channels, 234
Potential energy function, 191
Potential energy surface (PES), 191
Potential gradient, 274
Potential of mean force (PMF), 142, 147, 197, 270
Power series expansion, 266
PREDATOR, 110
Predicting protein structure, 9
Prepeptide, 89
Preprotein, 89
Presmoothing, 256
PRIDE, 25
Primary structure, 7, 90
Primitive Cartesian Gaussian basis functions, 314
Principle of microscopic reversibility, ix, 195
Principle of minimum frustration, 175
PRINTS, 39
PRISM, 25
Probability density functions (PDFs), 113, 275
Probability fluxes, 276
Probability tables, 140
Probable sequences, 67
Probable templates, 71, 82
PROCHECK, vii, 133, 138, 147
PRODOM, 39
PROF, 110
Profiles, 28, 66, 73, 91
Progressive alignment, 87, 95, 98
Prolongation, 256
ProSa, vii, 124, 142, 147, 153
PROSITE, vii, 39, 62
PROSUP, 25

Protein building blocks, 5
Protein chain thickness, 35
Protein channel, 231
Protein conformation space, 32
Protein conformations, 179
Protein crystallization, vi
Protein Data Bank (PDB), vii, 8, 13, 38, 63, 65, 68, 88, 120
Protein domain assignment, 16
Protein domain class, 40
Protein domains, vi, 12
Protein engineering, 216, 243
Protein fold space, viii
Protein folding, 61, 194
Protein folding class, 10
Protein folding mechanism, 171
Protein folding process, 170
Protein folding thermodynamics, 189
Protein function, 60
Protein gates, 229
Protein α-helix, 7, 58, 105, 187, 232
Protein Information Resource (PIR), 39, 65
Protein models, 179
Protein packing, 78
Protein relaxation times, 179
Protein Research Foundation (PRF), 65
Protein α-β sandwich, 187
Protein secondary structure, 73
Protein shape descriptors, 33, 35
Protein β-sheets, 7, 105, 187, 198, 232
Protein β-strands, 7, 90, 235
Protein structural domains, 15
Protein structure, 1, 4, 61, 170
Protein structure alignment programs, 25
Protein structure classifications, vi, 1, 35, 62
Protein structure comparisons, vi, 14, 35
Protein structure hierarchy, 5
Protein Structure Initiative (PSI), v
Protein structure resources, 13
Protein structure similarity, 14
Protein structure space, 44
Protein structure superposition, 23, 26
Protein transition states, 201
Protein unfolding, 195
Protein-nucleic acid complexes, 68
Proteins, v, 1, 231
α Proteins, 9, 40
ProtoMap, 39, 144
PROVE, vii, 138, 147
Pseudo-metric, 35
Pseudo-protein models, 74
PSI-PRED, 81, 110
3D-PSSM, 113, 119
PUU, 16, 42, 43
Pyramid algorithm, 304

Quantitative Structure Activity Relationship (QSAR), xi, 73, 296, 316
Quantitative Structure Property Relationship (QSPR), xi, 296, 316
Quantum chemistry, xi, 296, 314
Quaternions, 18
QuickSearch, 69

Radial distribution function (RDF), 268
Radius of curvature, 33, 217
Radius of gyration, 179
Ramachandran plots, 138, 139
Random coil, 90, 105
Random energy model (REM), 174
Random heteropolymers, 173
Random noise, 188
Random search through conformational space, 183
Rate-limiting step in protein folding, 189, 202
RCSB consortium, 13
Reaction coordinate, 176, 197, 202
Reaction models, 176
Real space, 245
Reciprocal space, 245
Reduced amino acid representations, 181
Reduced protein models, 187
Redundant transformations, 302
Reference force, 259
Refinement, 119, 124
Refolding process, 178
Regression, xi, 316, 320
Relational database, 14
Relaxation time, 180, 238
Replica methods, 174
Replica-exchange (REX), ix, 199, 200
Replica-exchange molecular dynamics (REMD), 200
Residue burial, 76
Residue pattern, 66
Residues, 5, 58
Resources for classification of protein sequences, 39
Restraining potential, 216
Restriction, 256
Retinal binding proteins, 116

Reverse position-specific BLAST (RPS-BLAST), 66
Reverse transform, 301
Ribosome, 89
Ridges, 302
Rigid-body transformation, 16
RMS/coverage plot, 33
Rotamer library, 121, 126
Rotamer searches, 125
Rough energy landscape, 173
Rugged energy landscape, 187

Salt bridges, 5
Sampling methods, 199
SAM-T02, 119
Sandwich topologies, 11
SARF2, 25
Satisfaction of Spatial Restraints, 113
Savitzky-Golay smoothing, 309, 311
Scaffold, 74, 76, 78, 119, 197
Scaled Gauss metric (SGM), 35
Scaling, 230
Scaling function, 303
Schrödinger equation, 314
Scientific classification, 2
Score, 95
Scoring functions, 26, 27
SearchStatus, 69
SearchFields, 69
SearchLite, 69
Secondary structure, 7, 41, 76, 79, 86, 89, 126, 132, 176, 200
Secondary structure elements (SSE), 8, 24, 40, 90, 118
Secondary structure prediction, 78, 110
Segment match modeling, 115, 116
Selecting templates, 104
Selectivity, 232
Selectivity filter, 234, 271
Self-consistency, 263, 279
Self-consistent simulation programs, 252
Self-force, 251
Sequence, 7, 10
Sequence alignment, 65, 70, 84
Sequence Alignment and Modeling (SAM), 70, 119
Sequence alignment methodologies, 86
Sequence identity, 81
Sequence similarity, 13, 57, 61, 73
Sequence to Coordinates (S2C) website, 88
Sequence-dependent thermodynamics, 183
Sequential folding, 172
SHAKE, 192
Shape descriptors, 33, 35
SHEBA, 26
Short time scales, 172
Short-range forces, x, 244, 258
Short-range interaction, 258, 271
Short-time Fourier transform (STFT), 298, 299
Side chains, 5, 105, 117, 119, 121
Side-chain conformational libraries, 126
Side-chain conformers, 137
Side-chain geometries, 125
Side-chain packing, 28
Side-Chains with Rotamer Library (SCWRL), vii, 125
Signal basis, 307
Signal characterization, 312
Signal cleaning, 296, 309
Signal components, 296
Signal compression, xi, 306, 313
Signal critical points, 312
Signal feature isolation, xi, 312
Signal information, 313
Signal noise, 309
Signal processing methods, xi, 295
Signal representation, 307
Signaling segment, 89
Silk, 7
Similarity, 14, 24, 29, 47, 66
Similarity matrix, 67, 86, 90, 91
Similarity measures, 43
Similarity score, 27, 32
Simple exact models, 181
Simple Modular Architecture Research Tool (SMART), 39, 67
Simple point charge (SPC) model, 269
Simulated annealing, 24, 125, 135
Simulated tempering, 180
Simulation box, 261
Simulation of protein folding, 169
Simulation techniques, 179
Single linkage clustering, 42, 47
Single template structure, 118
Single-domain proteins, 176
Singular value decomposition (SVD), 17, 18
Site-directed mutagenesis, 212
Size-dependent artifacts, 262
Skeletal models, 4
Skeleton wavelets, 302
Smoluchowski equation, 276
Smoothing, xi
Smoothing algorithm, 310

Solvation effects, 242
Solvation state, 271
Solvent, 267
Solvent accessibility, 28
Solvent exposure, 76
Solvent viscosity, 188
Solvent-accessible surface area (SASA), 78, 140, 192
Space scales, x, 241
Space-filling models, 4
Spatial inhomogeneities, 263
Spatial restraints, 113
SPC/E, 133, 269
Specialized proteomic databases, 80
Spectral compression, 313
Spectroscopy, xi
Sperm whale myoglobin, 58, 83, 85, 122, 151
Spherical harmonics, 246
Sphingolipids, 237
Spin glass systems, ix, 172, 174, 199
Spline wavelet, 305
Spurious relationships, 319
SRS, 13
SSAP, 26, 28, 42
SSM, 26
Statistically sound model, 60
Stereochemical assignments, 138
Stochastic difference equation (SDE), ix, 194
Stochastic separatrix, 204
Stopped-flow kinetics, 177
STRIDE notation, 71, 89
STRUCTAL, 29, 30
Structural biology, vi, 2
Structural classification methods, 38
Structural Classification of Proteins (SCOP), vi, 3, 32, 39, 40, 44
Structural domains, 14
Structural family, 120
Structural features, 86
Structural genomics projects, 3, 35
Structural molecular biology, 1
Structural relatedness, 40
Structural similarities, 16
Structural variance, 146
Structurally conserved regions (SCRs), 90, 118
Structurally variable regions (SVRs), 90, 118
Structure alignment, vi, vii
Structure databases, 1, 35
Structure of water, 239
Structure-based alignment, 90
Substructure, 23, 24
Successive overrelaxation (SOR) method, 254
Supercoiled DNA, 35
Superfamily, vi, 38, 40
Super-secondary structures, 7
Surface property distributions, 318
Surface Volume (SurVol), 144
Swiss Institute for Bioinformatics, 62
SWISS-MODEL, vii, 62, 119
SWISS-PROT, vii, 62, 65
Symmlet wavelet, 305
Systematic classifications, 2
SYSTERS, 39

Target, 58, 59, 67, 84, 116
Target protein, 68
Target sequence, 90
Target-template alignments, 80, 122
Taylor expansion, 246, 275
T-Coffee, vii, 99, 102, 112
Temperature, 200
Template, 57, 58, 59, 84, 90, 116
Template protein, 88
Template selection, 122
Template structure, 104
Temporal information, 297
Tertiary structure, 7, 90, 101, 115, 140, 176
Theoretical protein models, 68
Thermodynamic equilibrium, 179
Thermodynamic ϕ-values, 213
THREADER, 78, 81
Threading, vii, 71, 73
Threading algorithm, 79
Threading Expert, 81
Threading Onion Model (THOM), 79
Three-state model, 176, 189
TIGRFAMS, 39
TIM (triose phosphate isomerase), 12
TIM barrel, 12
Time domain, 297
Time scale, x, 172, 189, 192, 232, 240, 241
Time step, 192, 265
Time-dependent wave function, 315
Tinker, 124, 133
TIP3P, 133, 198
TIP4P, 133
TIP4P-Ew, 133
TIP4P-FQ, 273
TOPS, 10, 13, 26
TOPSCAN, 26
Training sets, 72
Transferable atom equivalent (TAE) descriptors, 317

Transferable intermolecular potential functions (TIPS), 269
Transformed wavelet, 301, 302
Transition path sampling, 210, 212
Transition state, ix, 199, 201, 202, 206, 211, 216
Transition state ensemble (TSE), ix, 201
Transition state theory (TST), 202
Translation variable, 303
Transmembrane helices, 234
Transport equations, 276
Tree of protein fragments, 14
TrEMBL, 62
Triangular inequality, 32
Triangular-shaped-cloud (TSC) charge, 251
TRIBES, 39
Triple-zeta basis sets, 314
Tsallis ensemble, 180
Turn regions, 41
Two-dimensional models, 182
Two-grid iteration, 256
Two-hit search method, 67
Two-state folders, 177, 201, 212
Two-state folding, 183
Two-state kinetics, ix
Two-state models, 176
Type II diabetes, 219

UCSF Chimera, 85, 112, 138
Ultra violet circular dichroism (UVCD), 176
Ultraviolet-visible spectroscopy, 311, 313
Umbrella sampling, 212, 271
UNDERTAKER, 70
Unfolded state, 174, 175, 176, 200
Unfolding rate, 178, 206
Unfolding trajectories, ix, 195
Unfolding transition states, 206
UniProt, 39, 62
Unlike contacts, 184
Unsuitable geometries, 142

Valence regions, 314
van der Waals forces, 258
van der Waals surface, 241
van't Hoff derived enthalpies, 176
Variable regions, 116, 121, 137
Variable target function method (VTFM), 115
Vassiliev knot invariants, 35
VAST, 26
Verify3D, vii, 124, 138, 140, 147, 151
Verlet algorithm, 188, 267
Verlet integration, 266, 269
Viruses, 68
Visualization, 4
VMD, 4, 138, 233, 236
Voltage-activated gate, 235
Voltage-sensor paddle, 235
Voltammetry, 311
Voronoi method, 144

Wang Landau method, 180
Washington University-BLAST (WU-BLAST), 68, 71
Water, 192, 263, 267
Water models, 268, 273
Water transport, 233
Wave function, 314
Wavelets, 295
Wavelet analysis, 305
Wavelet coefficient descriptors (WCDs), 317
Wavelet coefficients, 305, 310
Wavelet compression, 313
Wavelet families, 305
Wavelet function, 300, 303
Wavelet neural network (WNN), 313
Wavelet packet transform (WPT), 307
Wavelet selection, 306
Wavelet space, 296, 302, 303, 318
Wavelet thresholding, 310
Wavelet transform (WT), x, 295, 300
Weighted histogram analysis method (WHAM), 180, 181, 187, 190
Weighted superpositions, 21
Writhe, 33

X-ray absorption, 313
X-ray crystallography, 9, 23, 68, 144, 146
X-ray spectroscopy, 234

Z-scores, 43, 76, 81, 143