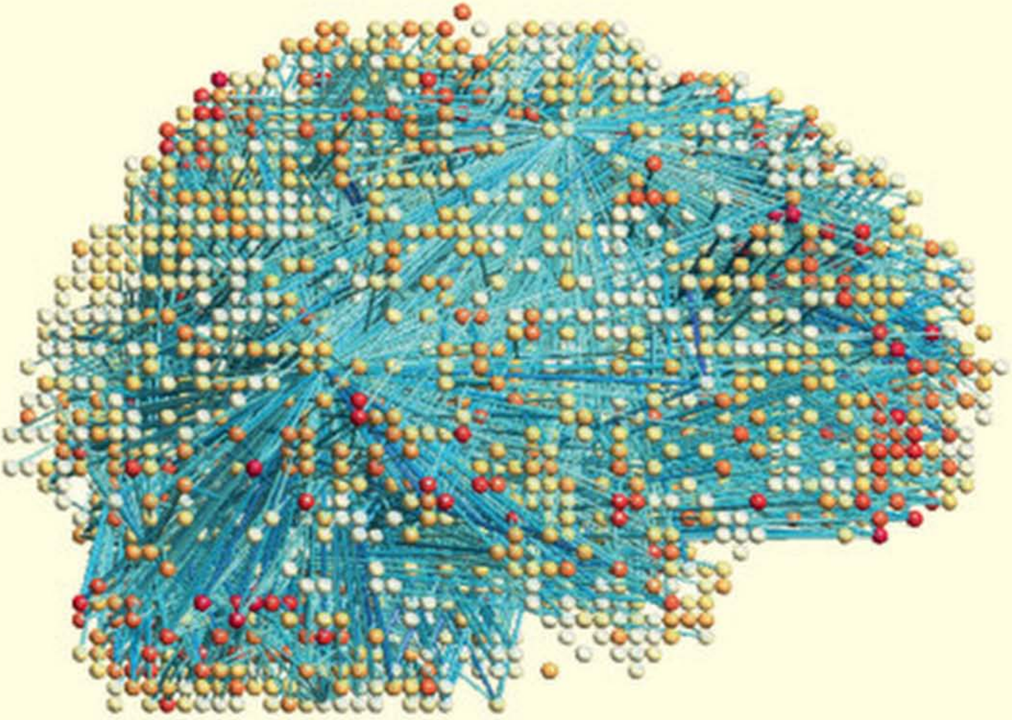
BRAIN NETWORK ANALYSIS



Moo K. Chung

Brain Network Analysis

This tutorial reference serves as a coherent overview of various statistical and mathematical approaches used in brain network analysis, where modeling the complex structures and functions of the human brain often poses many unique computational and statistical challenges. This book fills a gap as a textbook for graduate students while simultaneously articulating important and technically challenging topics. Whereas most available books are graph theory centric, this text introduces techniques arising from graph theory and expands to include other advanced models in its discussion on network science, regression, and algebraic topology. Links are included to the sample data and codes used in generating the book's results and figures, helping empower methodological understanding in a manner immediately usable to both researchers and students.

MOO K. CHUNG is an Associate Professor in the Department of Biostatistics and Medical Informatics at the University of Wisconsin–Madison and is also affiliated with the Department of Statistics and Waisman Laboratory for Brain Imaging and Behavior. He has received the Vilas Associate Award for his research in applied topology to medical imaging, the Editor's Award for best paper published in the *Journal of Speech, Language, and Hearing Research* for a paper that analyzed computed tomography (CT) images, and a National Institutes of Health (NIH) Brain Initiative Award for work on persistent homological brain network analysis. He has written numerous papers in computational neuroimaging and two previous books on computation on brain image analysis.

Brain Network Analysis

MOO K. CHUNG
University of Wisconsin–Madison

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India

79 Anson Road, #06-04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107184862
DOI: 10.1017/9781316882610

First published 2019

Printed in the United Kingdom by TJ International Ltd., Padstow, Cornwall

A catalogue record for this publication is available from the British Library.

Library of Congress Cataloging-in-Publication Data
Names: Chung, Moo K., author.
Title: Brain network analysis / Moo K. Chung.
Description: Cambridge, United Kingdom ; New York, NY : Cambridge University Press, 2019. | Includes bibliographical references and index.
Identifiers: LCCN 2019000330 | ISBN 9781107184862 (hardback : alk. paper)
Subjects: | MESH: Connectome–methods | Brain–physiology | Nerve Net–physiology | Neural Pathways–physiology | Models, Neurological | Mathematical Computing
Classification: LCC QP376 | NLM WL 141.5.B6 | DDC 612.8/2–dc23
LC record available at https://lccn.loc.gov/2019000330

ISBN 978-1-107-18486-2 Hardback

To my parents
for their endless love

Contents

Preface

Brain network analysis is an emerging field that utilizes various noninvasive brain imaging modalities such as magnetic resonance imaging (MRI), functional MRI (fMRI), positron emission tomography (PET), diffusion tensor imaging (DTI), and electroencephalography (EEG) in mapping out the four-dimensional (4D) spatiotemporal dynamics of the human brain networks in both normal and clinical populations at the macroscopic level. There has been substantial progress in the past decade on this topic. A major challenge in the field is caused by the massive amount of nonstandard high-dimensional network data that are difficult to analyze using available standard techniques. This requires new computational approaches and solutions.

The main goals of this book are to provide a coherent overview of various statistical and mathematical approaches used in brain network analysis to a wide range of researchers and students, and to articulate important yet technically challenging topics further. It is hoped that the book presents the coherent mathematical treatment of underlying methods. The book is mainly focused on methodological issues beyond widely used graph theory–based approaches. We wish to provide methodological understanding in a manner immediately usable to researchers and students. Concepts and methods are illustrated with brain imaging applications and examples. Some of the brain network data sets along with `MATLAB` and `R` codes used in the book can be downloaded from the author's website. The web links are provided in appropriate places. By making some of the data and codes available, we tried to make the book more accessible to a wide range of readers.

Although I am indebted to many colleagues and students in writing this book, I would particularly like to thank the following individuals, in no particular order. Richard Davidson, Andrew Alexander, Seth Pollak, Hill Goldsmith of the University of Wisconsin–Madison; David Zald of Vanderbilt University; and Benjamin Lahey of the University of Chicago provided various brain imaging data used in illustrating the methods. Hyekyoung Lee of Seoul National University and Yuan Wang of University of South Carolina helped me write chapters related to persistent homology and topological distances. Hernando Ombao of the King Abdullah University of Science and Technology and Dustin Pluta of the University of California–Irvine helped me write chapters related to the dynamic network models. Andrey Gritsenko of the University of Wisconsin–Madison performed some of basic image processing on the resting-state fMRI from Human Connectome Project data and helped compile the list of Automatic Anatomical Labeling (AAL) parcellation. Although most figures are produced by myself using `MATLAB`, some figures are generated by my current and former students, postdocs, and colleagues. Such figures are identified in figure captions and the proper credits are given. I am also indebted to Fred Boehm of the University of Wisconsin–Madison and Feng Liu of Harvard University for proofreading a few chapters.

1

Statistical Preliminary

This chapter covers the basic statistical methods that are mostly used in univariate voxel-level approaches. However, these basic methods are equally useful in brain network analysis as well. Most of network modeling techniques are based on the voxel-level methods. Readers familiar with univariate statistical methods can skip this chapter.

1.1 General Linear Models

General linear models (GLM) have been widely used in brain imaging and network studies. The GLM is a very flexible and general statistical framework encompassing a wide variety of fixed-effect models such as multiple regressions, the analysis of variance (ANOVA), the multivariate analysis of variance (MANOVA), the analysis of covariance (ANCOVA), and the multivariate analysis of covariance (MANCOVA) (Timm and Mieczkowski, 1997). More complex multilevel or hierarchical models such as the mixed-effects models and structural equation models (SEM) are also viewed as special cases of general linear models.

GLM provides a framework for testing various associations and hypotheses while accounting for nuisance covariates in the model in a straightforward fashion. The effect of age, sex, brain size, and possibly IQ may have severe confounding effects on the final outcome of many brain network studies. Older populations' reduced functional activation could be the consequence of age-related atrophy of neural systems (Mather et al., 2004). Brain volumes are significantly larger for children with autism 12 years old and younger compared with normally developing children (Aylward et al., 1999). Therefore, it is desirable to account for various confounding factors such as age and sex. This can be done using GLM automatically. The parameters of GLM are mainly estimated by the least squares estimation and have been implemented in many statistical packages such as R[1] (Pinehiro and Bates, 2002), statistical parametric mapping (SPM)[2] and fMRI-STAT.[3]

We assume there are n subjects. Let y_i be the response variable at a node or edge, which is mainly coming from images and $\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})$ to be the variables of interest and $\mathbf{z}_i = (z_{i1}, \cdots, z_{ik})$ to be nuisance variables corresponding to the ith subject. Then we have GLM

$$y_i = \mathbf{z}_i \boldsymbol{\lambda} + \mathbf{x}_i \boldsymbol{\beta} + \epsilon_i,$$

where $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_k)^\top$ and $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)^\top$ are unknown parameter vectors to be estimated. We assume ϵ to be the usual zero mean Gaussian noise.

The significance of the variable of interests $\mathbf{x}_i$ is determined by testing the null hypothesis

$$H_0 : \boldsymbol{\beta} = 0 \text{ vs. } H_1 : \boldsymbol{\beta} \neq 0.$$

The fit of the reduced model corresponding to $\beta = 0$, i.e.,

$$y_i = \mathbf{z}_i \boldsymbol{\lambda}, \tag{1.1}$$

is measured by the sum of the squared errors (SSE):

$$\text{SSE}_0 = \sum_{i=1}^{n} (y_i - \mathbf{z}_i \widehat{\boldsymbol{\lambda}}_0)^2,$$

where $\widehat{\boldsymbol{\lambda}_0}$ is the least squares estimation obtained from the reduced model. The reduced model (1.1) can be written in a matrix form

$$\underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}_{\mathbf{y}} = \underbrace{\begin{pmatrix} z_{11} & \cdots & z_{1k} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nk} \end{pmatrix}}_{\mathbf{Z}} \underbrace{\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix}}_{\boldsymbol{\lambda}}.$$

By multiplying $\mathbf{Z}^\top$ on the both sides, we obtain

$$\mathbf{Z}^\top \mathbf{y} = \mathbf{Z}^\top \mathbf{Z} \boldsymbol{\lambda}.$$

Now the matrix $\mathbf{Z}^\top \mathbf{Z}$ is a full rank and can be invertible if $n \geq k$, i.e., there are more subjects than the number of parameters. The matrix equation then can be solved by performing a matrix inversion

$$\widehat{\boldsymbol{\lambda}}_0 = (\mathbf{Z}^\top \mathbf{Z})^{-1} \mathbf{Z}^\top \mathbf{y}.$$

[1] www.r-project.org
[2] www.fil.ion.ucl.ac.uk/spm
[3] www.math.mcgill.ca/keith/fmristat

Similarly the fit of the full model corresponding to $\boldsymbol{\beta} \neq 0$, i.e.,

$$y_i = \mathbf{z}_i \boldsymbol{\lambda} + \mathbf{x}_i \boldsymbol{\beta}$$

is measured by

$$\mathrm{SSE}_1 = \sum_{i=1}^{n} (y_i - \mathbf{z}_i \widehat{\boldsymbol{\lambda}}_1 - \mathbf{x}_i \widehat{\boldsymbol{\beta}}_1)^2,$$

where $\widehat{\boldsymbol{\lambda}}_1$ and $\widehat{\boldsymbol{\beta}}_1$ are the least squares estimation from the full model. The full model can be written in a matrix form by concatenating the row vectors $\mathbf{z}_i$ and $\mathbf{x}_i$ into a larger row vector $(\mathbf{z}_i, \mathbf{x}_i)$, and the column vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$ into a larger column vector $(\boldsymbol{\lambda}^\top, \boldsymbol{\beta}^\top)^\top$, i.e.,

$$y_i = (\mathbf{z}_i, \mathbf{x}_i) \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\beta} \end{pmatrix}.$$

Then the parameters of the full model can be estimated in the least squares fashion. Note that

$$\begin{aligned} \mathrm{SSE}_1 &= \min_{\boldsymbol{\lambda}_1, \boldsymbol{\beta}_1} \sum_{i=1}^{n} (y_i - \mathbf{z}_i \boldsymbol{\lambda}_1 - \mathbf{x}_i \boldsymbol{\beta}_1)^2 \\ &\leq \min_{\boldsymbol{\lambda}_0} \sum_{i=1}^{n} (y_i - \mathbf{z}_i \boldsymbol{\lambda}_0)^2 = \mathrm{SSE}_0. \end{aligned}$$

So the larger the value of $\mathrm{SSE}_0 - \mathrm{SSE}_1$, more significant the contribution of the coefficients $\boldsymbol{\beta}$ is. Under the assumption of the null hypothesis H_0, the test statistic is the ratio

$$F = \frac{(\mathrm{SSE}_0 - \mathrm{SSE}_1)/p}{\mathrm{SSE}_0/(n-p-k)} \sim F_{p,n-p-k}. \tag{1.2}$$

The larger the F value, it is more unlikely to accept H_0.

1.1.1 *T*-Statistic

When $p = 1$, the test statistic F is distributed as $F_{1,n-1-k}$, which is the square of the student t-distribution with $n - 1 - k$ degrees of freedom, i.e., t^2_{n-1-k}. In this case, it is better to use t-statistic. The advantage of using the t-statistic is that the test statistic can provide the direction of the group difference that the F-statistic cannot provide.

Let

$$c = (\underbrace{0, \cdots, 0}_{k}, 1, \underbrace{0, \cdots, 0}_{p-1})^\top$$

be the contrast vector of size $k + p$. The incorporation of the contrast vector makes the algebraic derivation straightforward. Consider testing the significance of $H_0 : \beta_1 = 0$. The least squares estimation of β_1 can be written as

$$\widehat{\beta_1} = c \begin{pmatrix} \widehat{\boldsymbol{\lambda}} \\ \widehat{\boldsymbol{\beta}} \end{pmatrix}.$$

Under the assumption $\epsilon_i \sim N(0, \sigma^2)$,

$$\mathbb{E}\widehat{\beta_1} = \beta_1.$$

Further, the variance

$$\mathbb{V}\widehat{\beta_1} = c\mathbb{V} \begin{pmatrix} \widehat{\boldsymbol{\lambda}} \\ \widehat{\boldsymbol{\beta}} \end{pmatrix} c^\top = \sigma^2 c^\top \left([\mathbf{ZX}]^\top \mathbf{ZX} \right)^{-1} c.$$

Thus, the unbiased estimator of σ^2 is given by

$$\mathrm{SSE}_1/(n - 1 - k).$$

We plug this estimator into σ^2. Then the test statistic under the null hypothesis is

$$T = \frac{\widehat{\beta_1}}{\sqrt{\mathbb{V}\widehat{\beta_1}}} \sim t_{n-1-k}.$$

1.1.2 R-Square

The R-square of a model explains the proportion of variability in measurement that is accounted by the model. Sometime R-square is called the coefficient of determination and it is given as the square of a correlation coefficient for a very simple model. For a linear model involving the response variable y_i, the total sum of squares (SST) measures total total variation in response y_i and is defined as

$$\mathrm{SST} = \sum_{i=1}^{n} (y_i - \bar{y})^2,$$

where $\bar{y}$ is the sample mean of y_i.

On the other hand, SSE measures the amount of variability in y_i that is not explained by the model. Note that SSE is the minimum of the sum of squared residual of any linear model, SSE is always smaller than SST. Therefore, the amount of variability explained by the model is SST-SSE. The proportion of variability explained by the model is then

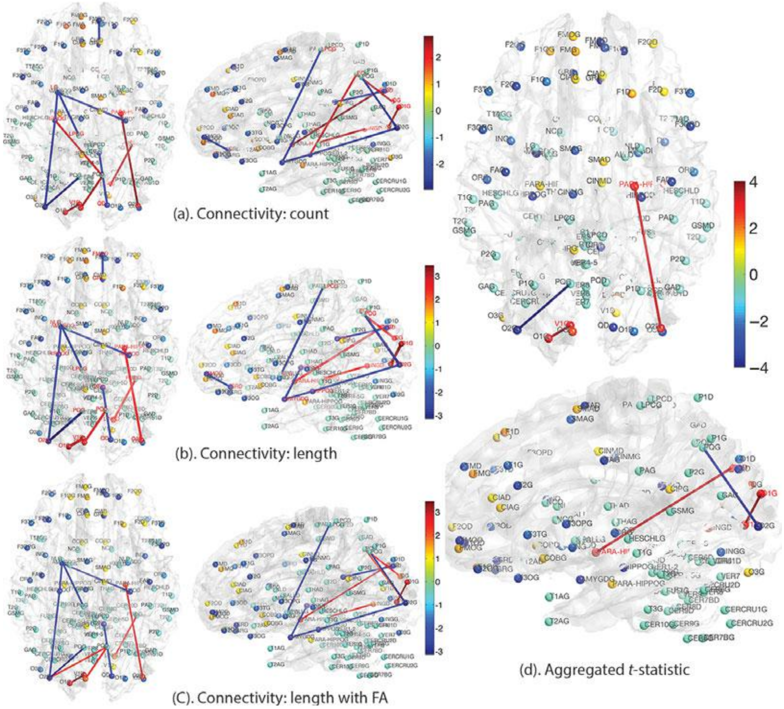
$$R^2 = \frac{\mathrm{SST} - \mathrm{SSE}}{\mathrm{SST}},$$

which is the coefficient of determination. The R-square ranges between 0 and 1 and the value larger than 0.5 is usually considered significant.

1.1.3 Sum of T-Statistics

Often there is a situation such as a meta-analysis, where we have to sum the t-statistic images or networks (Chung et al., 2017b). Note that a t-statistic for large degrees of freedom (above 30) is very close to standard normal, i.e., $N(0,1)$. For n identically distributed possibly dependent t-statistics $t^1, \cdots, t^n$, the variance of sum $\sum_{j=1}^{n} t^j$ is approximately given by (Billingsley, 1995)

$$\mathbb{V}\left(\sum_{j=1}^{n} t^j\right) \approx n + \sum_{i \neq j} \mathbb{E}(t^i t^j),$$



(a). Connectivity: count

(b). Connectivity: length

(C). Connectivity: length with FA

(d). Aggregated t-statistic

Figure 1.1 (a)–(c) t-statistic results of group difference between maltreated children and normal controls for three different connectivity methods (Chung et al., 2017b). Only the connections at the p-value less than 0.01 (uncorrected) are shown. (d) The three t-statistic maps are aggregated to form a single t-statistic.

where $\mathbb{E}(t^i t^j)$ is the correlation between t^i and t^j. We used the fact $\mathbb{E}t^j = 0$. Then, we have the *aggregated t-statistic* given by

$$T = \frac{\sum_{j=1}^{n} t^j}{\sqrt{n + \sum_{i \neq j} \mathbb{E}(t^i t^j)}} \sim N(0,1).$$

If the statistics t^j are all independent, since t^j are close to standard normal, $\mathbb{E}(t^i t^j) \approx 0$. The dependency increases the variance estimate and reduces the aggregated t-statistic value. Unfortunately, it is difficult to estimate the correlations directly since only one t-statistic map is available for each t^j. $\mathbb{E}(t^i t^j)$ can be empirically estimated by computing correlations over the entries of t-statistic maps t^i and t^j (see Figure 1.1).

1.2 Logistic Regression

Logistic regression is useful for setting up a probabilistic model on the strength of connectivity and performing classification (Subasi and Ercelebi, 2005). Suppose k regressors $X_1, \cdots, X_k$ are given. These are both imaging and nonimaging biomarkers such as gender, age, education level, and memory test score. Let $x_{i1}, \cdots, x_{ik}$ denote the measurements for the ith subject. Let the response variable Y_i be the probability of connection modeled as a Bernoulli random variable with parameter π_i, i.e.,

$$Y_i \sim \text{Bernoulli}(\pi_i).$$

$Y_i = 0, 1$ indicates the edge connected (assigned number 1) or disconnected (assigned number 0) respectively. π_i is then the likelihood (probability) of the edge connected, i.e., $\pi_i = P(Y_i = 1)$.

Now consider linear model

$$Y_i = \mathbf{x}_i^\top \beta + \epsilon_i, \tag{1.3}$$

where $\mathbf{x}_i^\top = (1, x_{i1}, \cdots, x_{ik})$ and $\beta^\top = (\beta_0, \cdots, \beta_k)$. We may assume

$$\mathbb{E}\epsilon_i = 0, \quad \mathbb{V}\epsilon_j = \sigma^2.$$

However, linear model (1.3) is no longer appropriate since

$$\mathbb{E}Y_j = \pi_i = \mathbf{x}_i^\top \beta$$

but $\mathbf{x}_i^\top \beta$ may not be in the range $[0, 1]$. The inconsistency is caused by trying to match continuous variables x_{ij} to categorical variable Y_i directly. To address this problem, we introduce the *logistic regression function* g:

$$\pi_i = g(x_i) = \frac{\exp(\mathbf{x}_i^\top \beta_i)}{1+\exp(\mathbf{x}_i^\top \beta_i)}. \tag{1.4}$$

Using the *logit function*, we can write (1.4) as

$$\operatorname{logit}(\pi_i) = \log \frac{\pi_i}{1-\pi_i} = \mathbf{x}_i^\top \beta_i.$$

1.2.1 Maximum Likelihood Estimation

The unknown parameters β are estimated via the maximum likelihood estimation (MLE) over n subjects at each edge. The likelihood function is

$$\begin{aligned} L(\beta|y_1,\cdots,y_n) &= \prod_{i=1}^n \pi_i^{y_i}(1-\pi_i)^{1-y_i} \\ &= \prod_{i=1}^n \left[\frac{\exp(\mathbf{x}_i^\top \beta_i)}{1+\exp(\mathbf{x}_i^\top \beta_i)}\right]^{y_i} \prod_{i=1}^n \left[\frac{1}{1+\exp(\mathbf{x}_i^\top \beta)}\right]^{1-y_i}. \end{aligned}$$

The loglikelihood function is given by

$$\begin{aligned} \log L(\beta) &= \text{const.} + \sum_{i=1}^n y_i \log \pi_i + (1-y_i)\log(1-\pi_i) \\ &= \text{const.} + \sum_{i=1}^n y_i \mathbf{x}_i^\top \beta + \log(1-\pi_i) \end{aligned}$$

and its maximum is obtained when

$$\frac{\partial \log L(\beta)}{\partial \beta} = \sum_{i=1}^n \mathbf{x}_i(y_i - \pi_i) = 0.$$

In simplifying the expression, we used the following identities

$$\frac{\partial \pi_i}{\partial \beta_0} = \pi_i(1-\pi_i)$$

and

$$\frac{\partial \pi_i}{\partial \beta_1} = x_i\pi_i(1-\pi_i).$$

Since the logistic regression function π is in complicated form, the maximum is obtained numerically. Define the *information matrix* $I(\beta)$ to be

$$I(\beta) = -\frac{\partial^2 \log L(\beta)}{\partial \beta' \partial \beta} - \sum_{i=1}^n \pi_i(1-\pi_i)\mathbf{x}_i\mathbf{x}_i^\top.$$

Then the Newton–Raphson algorithm is used to find the MLE in an iterative fashion. Starting with an arbitrary initial vector β^0, we estimate iteratively

$$\beta^{j+1} = \beta^j + I(\beta^j)^{-1}\frac{\partial \log L(\beta)}{\partial \beta}(\beta^j).$$

Many computational packages such as R and MATLAB have the logistic regression model fitting procedure.

Although we do not have the explicit formulas for the MLE, using the asymptotic normality of the MLE, the distributions of the estimators can be approximately determined. For large sample size n, the distribution of $\widehat{\beta}$ is approximately multivariate normal with means β with the covariance matrix $I(\widehat{\beta})^{-1}$.

1.2.2 Best Model Selection

Consider following full model:

$$\text{logit}(\pi_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p.$$

Let $\beta^{(1)} = (\beta_0, \cdots, \beta_q)^\top$ and $\beta^{(2)} = (\beta_{q+1}, \cdots, \beta_p)^\top$. The parameter $\beta^{(1)}$ corresponds to the parameters of the *reduced model*. Then we are interested in testing

$$H_0 : \beta^{(2)} = 0.$$

Define the *deviance* D of a model as $D = -2\log L(\widehat{\pi})$, which is distributed asymptotically as χ^2_{n-p-1}. Let $\widehat{\pi}^{(p)}$ and $\widehat{\pi}^{(q)}$ be the estimated success probabilities for the full and reduced models, and let D_p and D_q be the associated deviances. Then the log-likelihood ratio statistic for testing $\beta^{(2)} = 0$ is

$$2[\log L(\widehat{\pi}^{(p)}) - \log L(\widehat{\pi}^{(q)})] = D_q - D_p \sim \chi^2_{p-q}.$$

1.2.3 Logistic Discriminant Analysis

Discriminant analysis resulting from the estimated logistic model is called the *logistic discrimination*. We classify the ith subject according to a *classification rule*. The simplest rule is to assign the ith subject as group 1:

$$P(Y_i = 1) > P(Y_i = 0).$$

This statement is equivalent to $\pi_i > 1/2$. Depending on the bias and the error of the estimation, the value $1/2$ can be adjusted. For the fitted logistic model, we classify the ith subject as group 1 if $\mathbf{x}_i^\top \beta_i > 0$ and as 0 if $\mathbf{x}_i^\top \beta_i < 0$. The plane $\mathbf{x}_i^\top \beta = 0$ is the *classification boundary* that separates two groups.

The performance of classification technique is measured by the *error rate* γ, the overall probability of misclassification. The *cross-validation* is used to estimate the error rate. This is done by randomly partitioning the data into the training and the testing sets. In the *leave-one-out* scheme, the training set consists of $n-1$ subjects, while the testing set consists of one subject. Suppose the ith subject is taken as the test set. Then using the training set, we determine the logistic model. Using the predicted model, we test if the ith subject is correctly classified. The error rate obtained in this fashion is denoted as e_{-i}. Note that $e_{-i} = 0$ if the subject is classified correctly while $e_{-i} = 1$ if the subject is misclassified. The *leave-one-out error rate* is then given by

$$\widehat{\gamma} = \frac{1}{n}\sum_{i=1}^{n} e_{-i}.$$

To formally test the statistical significance of the discriminant power, we use Press's Q statistic (Hair et al., 1998), which is given by

$$n(2\gamma - 1)^2 \sim \chi_1^2.$$

Press's Q statistic is asymptotically distributed as χ^2 with one degree of freedom.

1.3 Random Fields

At the voxel level, it is often necessary to model measurements at each voxel as a random field. For instance, the deformation field of warping a brain to another brain is often modeled as a continuous random field (Chung et al., 2001b). The generalization of a continuous stochastic process defined in $\mathbb{R}$ to a higher dimensional abstract space is called a *random field*. For an introduction to random fields, see (Yaglom, 1987; Dougherty, 1999; Adler and Taylor, 2007). In the random field theory as introduced in (Worsley, 1994; Worsley et al., 1996b), measurement Y at voxel position $x \in \mathcal{M}$ is modeled as

$$Y(x) = \mu(x) + \epsilon(x),$$

where μ is the unknown functional signal to be estimated and ϵ is the measurement error, which is modeled as a random variable at each fixed x. Then the collection of random variables $\{\epsilon(x) : x \in \mathcal{M}\}$ is called a *stochastic process* or *random field*. The more precise measure-theoretic definition can be found in (Adler and Taylor, 2007). Random field modeling can be done beyond the usual Euclidean space to curved cortical and subcortical manifolds (Joshi, 1998; Chung et al., 2003a). Most of concepts in random fields are the continuous generalization of random vectors.

Definition 1.1 *Given a probability space, a random field $T(x)$ defined in $\mathbb{R}^n$ is a function such that for every fixed $x \in \mathbb{R}^n$, $T(x)$ is a random variable on the probability space.*

Definition 1.2 *The covariance function $R(x, y)$ of a random field T is defined as*

$$R(x, y) = \mathbb{E}\big[T(x) - \mathbb{E}T(x)\big]\big[T(y) - \mathbb{E}T(y)\big].$$

If the joint distribution of T at points $x_1, \cdots, x_m$

$$P\Big(T(x_1) \le z_1, \cdots, T(x_m) \le z_m\Big)$$

is invariant under the translation

$$(x_1, \cdots, x_m) \to (x_1 + \tau, \cdots, x_m + \tau),$$

T is said to be stationary or homogeneous.

For a stationary random field T, its covariance function is

$$R(x, y) = f(x - y)$$

for some function f. A special case of stationary fields is an *isotropic* field, which requires the covariance function to be rotation invariant, i.e.,

$$R(x, y) = f(|x - y|)$$

for some function f (Yaglom, 1987).

1.3.1 Gaussian Fields

The most important class of random fields is Gaussian fields. A more rigorous treatment can be found in Adler and Taylor (2007). Let us start defining a multivariate normal distribution from a Gaussian random variable.

Definition 1.3 *A random vector $T = (T_1, \cdots, T_m)$ is multivariate normal if $\sum_{i=1}^m c_i T_i$ is Gaussian for every possible $c_i \in \mathbb{R}$.*

Then a Gaussian random field can be defined from a multivariate normal distribution.

Definition 1.4 *A random field T is a* Gaussian random field *if $T(x_1), \cdots, T(x_m)$ are multivariate normal for every $(x_1, \cdots, x_m) \in \mathbb{R}^m$.*

An equivalent definition to Definition 1.4 is as follows. T is a Gaussian random field if the finite joint distribution

$$P(T(x_1) \le z_1, \cdots, T(x_m) \le z_m)$$

is a multivariate normal for every $(x_1, \cdots, x_m)$.

T is a mean zero Gaussian field if $\mathbb{E}T(x) = 0$ for all x. Because any mean zero multivariate normal distribution can be completely characterized by its covariance matrix, a mean zero Gaussian random field T can be similarly determined by its covariance function R. Two fields T and S are independent if $T(x)$ and $S(y)$ are independent for every x and y. For mean zero Gaussian fields T and S, they are independent if and only if the covariance function

$$R(x, y) = \mathbb{E}\big[T(x)T(y)\big]$$

vanishes for all x and y, which is a very strong assumption.

The Gaussian white noise is a Gaussian random field with the Dirac-delta function δ as the covariance function. Note the Dirac delta function is defined as $\delta(x) = \infty$ if $x = 0$ and $\delta(x) = 0$ if $x \neq 0$. Further, $\int \delta(x) = 1$. Numerically we can simulate the Dirac-delta function as the limit of the sequence of Gaussian kernel K_σ when $\sigma \to \infty$. The Gaussian white noise is simulated as an independent and identical Gaussian random variable at each voxel.

1.3.2 Derivative of Gaussian Fields

Suppose $\mathcal{G}$ is a collection of Gaussian random fields. For given $X, Y \in \mathcal{G}$, we have $c_1X + c_2Y \in \mathcal{G}$ again for all c_1 and c_2. Therefore, $\mathcal{G}$ forms an infinite-dimensional vector space. Any linear combination of Gaussian fields is again a Gaussian field. We can show that the derivatives of Gaussian fields are also Gaussian. To see this, we define mean-square convergence.

Definition 1.5 *A sequence of random fields T_h, indexed by h, converges to T as $h \to 0$ in mean-square if*

$$\lim_{h\to 0} \mathbb{E}\big|T_h - T\big|^2 = 0.$$

We will denote the mean-square convergence using the usual limit notation:

$$\lim_{h\to 0} T_h = T.$$

The convergence in mean square implies the convergence in mean. This can be seen from

$$\mathbb{E}\big|T_h - T\big|^2 = \mathbb{V}\big[T_h - T\big]^2 + \big(\mathbb{E}|T_h - T|\big)^2.$$

Now let $T_h \to T$ in mean square. Each term in the right-hand side should also converge to zero, proving the statement.

Now we define the derivative of the field in the mean-square sense as

$$\frac{dT(x)}{dx} = \lim_{h\to 0} \frac{T(x+h) - T(x)}{h}.$$

If $T(x)$ and $T(x+h)$ are Gaussian, $T(x+h) - T(x)$ is again Gaussian. Thus, the limit on the right-hand side is also Gaussian. If R is the covariance function of the mean zero Gaussian field T, the covariance function of its derivative field is given by

$$\mathbb{E}\Big[\frac{dT(x)}{dx}\frac{dT(y)}{dy}\Big] = \frac{\partial^2 R(x,y)}{\partial x \partial y}.$$

1.3.3 Integration of Gaussian Fields

The integration of Gaussian fields is also Gaussian. To see this, define the integration of a random field as the limit of Riemann sum. Let $\cup_{i=1}^{n}\mathcal{M}_i$ be a partition of $\mathcal{M}$, i.e.,

$$\mathcal{M} = \cup_{i=1}^{n}\mathcal{M}_i \text{ and } \mathcal{M}_i \cap \mathcal{M}_j = \emptyset \text{ if } i \neq j.$$

Let $x_i \in \mathcal{M}_i$ and $\mu(\mathcal{M}_i)$ be the volume of $\mathcal{M}_i$. Then we define the integration of field T as

$$\int_{\mathcal{M}} T(x)\, dx = \lim_{n\to\infty} \sum_{i=1}^{n} T(x_i)\mu(\mathcal{M}_i),$$

where the limit is taken as $n \to \infty$ and $\mu(\mathcal{M}_j) \to 0$ for all j. When we integrate a Gaussian field, it is the limit of a linear combination of Gaussian random variables so it is again a Gaussian random variable. In general, any linear operation on Gaussian fields will result in Gaussian fields with different covariance structures.

We can use a collection of Gaussian fields to construct χ^2-, t-, F-fields (Worsley, 1994; Worsley et al., 1996b, 2004; Cao and Worsley, 1999a). The χ^2-field with m degrees of freedom is defined as

$$T(x) = \sum_{i=1}^{m} X_i^2(x),$$

where $X_1, \cdots, X_m$ are independent, identically distributed Gaussian fields with zero mean and unit variance. Similarly, we can define t and F fields as well as Hotelling's T^2 field (Thompson et al., 1997; Collins et al., 1998; Joshi, 1998; Cao and Worsley, 1999a; Gaser et al., 1999).

1.3.4 Simulating Gaussian Fields

We show how to simulate smooth Gaussian fields by performing Gaussian kernel smoothing on white noise. This is perhaps the easiest way of simulating Gaussian fields.

White noise is defined as a random field whose covariance function is proportional to the Dirac-delta function δ, i.e.,

$$R(x, y) \propto \delta(x - y).$$

For instance, we may take

$$R(x, y) = \lim_{\sigma \to 0} K_\sigma(\|x - y\|),$$

the limit of the usual isotropic Gaussian kernel. White noise is usually characterized via generalized functions. One example of white noise is the generalized derivative of Brownian motion (Wiener process) called Gaussian white noise.

Definition 1.6 *Brownian motion (Wiener process) $B(x), x \in \mathbb{R}^+$ is a zero mean Gaussian field with covariance function*

$$R_B(x, y) = \min(x, y).$$

Following Definition 1.6, we have $\mathbb{V}B(x) = x$. The increments of Wiener processes in nonoverlapping intervals are independent identically distributed (i.i.d.) Gaussian. Further, the paths of the Wiener process are continous while they are not differentiable (Øksendal, 2010). Higher-dimensional Brownian motion can be generalized by taking each component of vector fields to be i.i.d. Brownian motion.

Although the path of the Wiener process is not differentiable, we can define the generalized derivative via integration by parts with a smooth function f called a test function in the following way

$$f(x)B(x) = \int_0^x f(y)\frac{dB(y)}{dy}\,dy + \int_0^x \frac{f(y)}{dy}B(y)\,dy.$$

Taking the expectation on both sides, we have

$$\int_0^x f(y)\mathbb{E}\frac{dB(y)}{dy}\,dy = 0.$$

It should be true for all smooth f so $\mathbb{E}\frac{dB(y)}{dy} = 0$. Further, it can be shown that the covariance function of process

$$dB(y)/dy \propto \delta(x - y).$$

The Gaussian white noise can be used to construct smooth Gaussian random fields of the form

$$X(x) = K * W(x) = K * \frac{dB(x)}{dx},$$

where K is a Gaussian kernel and W is the generalized derivative of Brownian motion. Since Brownian motion is a zero-mean Gaussian process, $X(x)$ is obviously a zero-mean field with the covariance function

$$\begin{aligned} R_X(x,y) &= \mathbb{E}[K * W(x) K * W(y)] && (1.5) \\ &\propto \int K(x-z)K(y-z)\,dz. && (1.6) \end{aligned}$$

The case when K is an isotropic Gaussian kernel was investigated by Siegmund and Worsley with respect to optimal filtering in scale space (Siegmund and Worsley, 1996).

In numerical implementation, we use the discrete white Gaussian noise, which is simply a Gaussian random variable.

Example 1.1 *Let w be a discrete version of white Gaussian noise given by*

$$w(x) = \sum_{i=1}^{m} Z_i \delta(x - x_i),$$

where i.i.d. $Z_i \sim N(0, \sigma_w^2)$. Note that

$$K * w(x) = \sum_{i=1}^{m} Z_i K(x - x_i). \tag{1.7}$$

The collection of random variables $K * w(y_1), \cdots, K * w(y_l)$ forms a multivariate normal at arbitrary points $y_1, \cdots, y_l$. Hence, the field $K * w(x)$ is a Gaussian field.

The covariance function of the field (1.7) is given by

$$\begin{aligned} R(x,y) &= \sum_{i,j=1}^{m} \mathbb{E}(Z_i Z_j) K(x - x_i) K(y - x_j) && (1.8) \\ &= \sum_{i=1}^{m} \sigma_w^2 K(x - x_i) K(y - x_i). && (1.9) \end{aligned}$$

As usual, we may take K to be a Gaussian kernel. Let us simulate some Gaussian fields.

Example 1.2 *Gaussian white noise is generated using $w \sim N(0, 0.4^2)$, which is shown in the top-left of Figure 1.2. With a Gaussian kernel with bandwidth 1, iteratively smoother versions of Gaussian random fields are constructed by*

```
K=inline('exp(-(x.^2+y.^2)/2/sigma^2)^{\top})
[dx,dy]=meshgrid([-10:10])
```

```
sigma=1
K=K(sigma,dx,dy)/sum(sum(K(sigma,dx,dy)))

w=normrnd(0,0.4,101,101)
smooth_w=w
for i=1:10
  figure; imagesc(smooth_w)
  smooth_w=conv2(smooth_w,K,'same')
end;
```

Figure 1.2 shows one, four, and nine iterations.

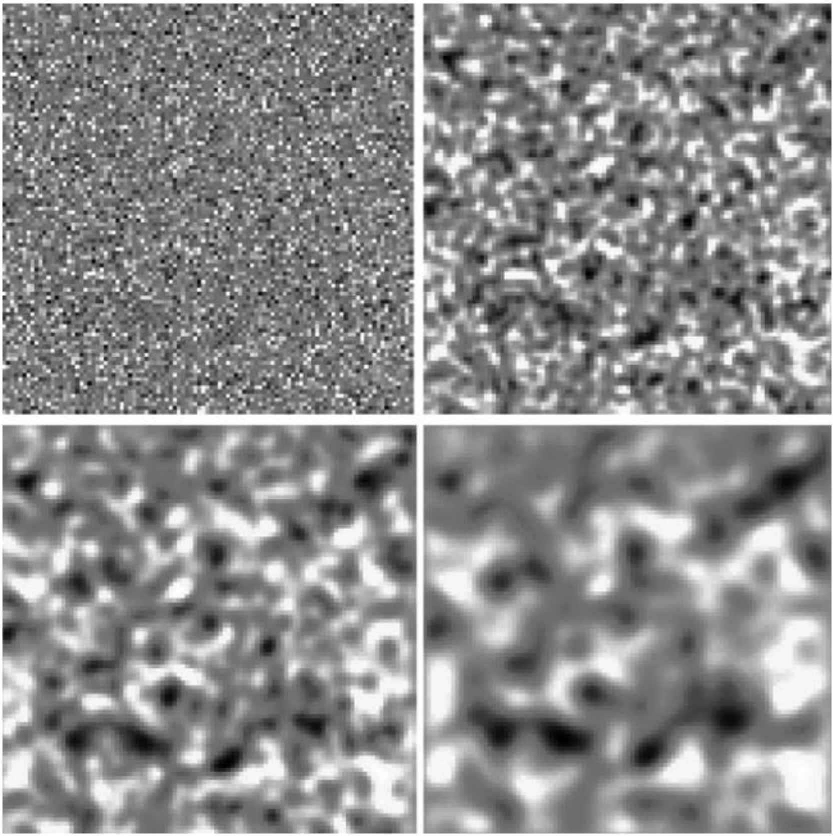


Figure 1.2 Gaussian random field simulation. Starting with Gaussian white noise $N(0, 0.4^2)$ (top-left), we iteratively apply Gaussian kernel smoothing one, four, and nine times with bandwidth $\sigma = 1$.

1.4 Statistical Inference on Fields

Given functional measurement Y, we have model

$$Y(x) = \mu(x) + \epsilon(x), \tag{1.10}$$

where μ is unknown signal and ϵ is a zero-mean unit variance Gaussian field (Worsley et al., 1996b; Miller et al., 1997; Joshi, 1998; Kiebel et al., 1999; Friston, 2002). We assume $x \in \mathcal{M} \subset \mathbb{R}^n$. The unknown signal is usually estimated by various spatial smoothing techniques over $\mathcal{M}$. The most widely used smoothing method is kernel smoothing and its variants because of their simplicity, and because they provide the theoretical basis for scale spaces and Gaussian random field theory (Worsley et al., 1995, 1996b).

In the usual SPM framework (Friston, 2002; Kiebel et al., 1999; Worsley et al., 1996b), inference on the model (1.10) proceeds as follows. If we denote an estimate of the signal by $\widehat{\mu}$, the residual $f - \widehat{\mu}$ gives an estimate of the noise. One then constructs a test statistic $T(x)$, corresponding to a given hypothesis about the signal. As a way to account for spatial correlation of the statistic $T(x)$, the global maximum of the test statistic over the search space $\mathcal{M}$ is taken as the subsequent test statistic. Hence a great deal of the neuroimaging and statistical literature has been devoted to determining the distribution of $\sup_{x\in\mathcal{M}} T(x)$ using random field theory (Worsley et al., 1996b; Taylor and Worsley, 2008), permutation tests (Nichols and Hayasaka, 2003) and the Hotelling–Weyl volume of tubes calculation (Naiman, 1990).

1.4.1 Type-I Error

In brain imaging and network analysis, one of the most important problems is that of signal detection, which can be stated as the problem of identifying the regions of statistical significance. So it can be formulated as an inference problem

$$H_0 : \mu(x) = 0 \text{ for all } x \in \mathcal{M}$$

vs.

$$H_1 : \mu(x) > 0 \text{ for some } x \in \mathcal{M}.$$

Let

$$H_0(x) : \mu(x) = 0$$

at a fixed point x. Then the null hypothesis H_0 is a collection of multiple hypotheses $H_0(x)$ over all x. Therefore, we have

$$H_0 = \bigcap_{x \in \mathcal{M}} H_0(x).$$

We may assume that $\mathcal{M}$ is the region of interest consisting of the finite number of voxels in a 3D volume and edges in network data. We also have the corresponding pointwise alternate hypothesis

$$H_1(x) : \mu(x) > 0$$

at each fixed point x. The alternate hypothesis H_1 is then constructed as

$$H_1 = \bigcup_{x \in \mathcal{M}} H_0(x).$$

If we use Z-statistic as a test statistic, for instance, we will reject each $H_0(x)$ if $Z > h$ for some threshold h. So at each fixed x, for level $\alpha = 0.05$ test, we need to have $h = 1.64$. However, if we threshold at $\alpha = 0.05$, 5% of observations are false positives. Note that the false positives are pixels where we are incorrectly rejecting $H_0(x)$ when it is actually true. However, these are the false positives related to testing $H_0(x)$. For determining the true false positives associated with testing H_0, we need to account for multiple comparisons.

Definition 1.7 *The type-I error is the probability of rejecting the null hypothesis (there is no signal) when the alternate hypothesis (there is a signal) is true.*

The type-I error, denoted as α, is also called the *familywise error rate* (FWER) and given by

$$\begin{aligned} \alpha &= P(\text{reject } H_0 \mid H_0 \text{ true }) \\ &= P(\text{ reject some } H_0(x) \mid H_0 \text{ true }) \\ &= P\Big(\bigcup_{x \in \mathcal{M}} \{Y(x) > h\} \Big| \mathbb{E}Y = 0 \Big). \end{aligned} \tag{1.11}$$

Given α level, we often find threshold h. Any voxels or edges that are above the threshold are considered a signal. Unfortunately, $Y(x)$ is correlated over x, and it makes the computation of the type-I error almost intractable for random fields other than Gaussian.

1.4.2 Bonferroni Correction

One standard method for dealing with multiple comparisons is to use the Bonferroni correction. Note that the probability measure is additive so that for any event E_j, we have

$$P\Big(\bigcup_{j=1}^{\infty} E_j\Big) \le \sum_{j=1}^{\infty} P(E_j).$$

This inequality is called the *Bonferroni inequality*, and it has been used in the construction of simultaneous confidence intervals and multiple comparisons when the number of hypotheses is small. From (1.11), we have

$$\alpha = P\Big(\bigcup_{x\in\mathcal{M}} \{Y(x) > h\} \,\Big|\, \mathbb{E}Y = 0\Big) \tag{1.12}$$

$$\le \sum_{x\in\mathcal{M}} P\big(Y(x_j) > h \mid \mathbb{E}Y = 0\big) \tag{1.13}$$

So by controlling each type-I error separately at

$$P\big(Y(x_j) > h \mid \mathbb{E}Y = 0\big) < \frac{\alpha}{\#\mathcal{M}}$$

we can construct the correct level α test. Here $\#\mathcal{M}$ is the number of voxels or edges, where the test statistic is defined.

The problem with the Bonferroni correction is that it is too conservative. The Bonferroni inequality (1.13) becomes exact when the measurements across voxels/edges are all independent, which is unrealistic. Since the measurements are expected to be strongly correlated across voxels/edges, we have highly correlated statistics. So in a sense, we have a smaller number of comparisons to make.

Consider 100×100 image Y of standard normal distributions (Figure 1.3). The threshold corresponding to the significance $\alpha = 0.05$ is 1.64. By thresholding the image at 1.64, we obtain approximately about 5% of pixels as false positives. To account for the false positives, we perform the Bonferroni correction. For an image of size 100×100, there are 10,000 pixels. Therefore,

$$\frac{\alpha}{\#\mathcal{M}} = \frac{0.05}{10000} = 0.000005$$

is the corresponding pointwise p-value and the corresponding threshold is 4.42. In this example, there is no pixel that is higher than 4.42 so we are not detecting any false positives as expected. In MATLAB, the threshold is found by

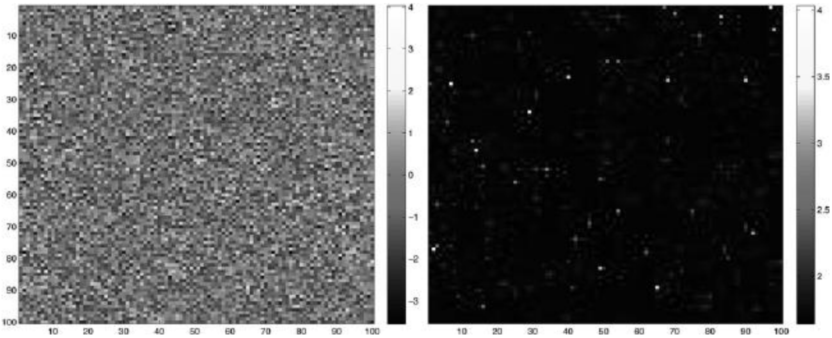


Figure 1.3 Image consisting of Gaussian white noise $N(0,1)$ at each pixel. At the thresholding 1.64 corresponding to the significance level $\alpha = 0.05$, 5% of all pixels are false positives.

```
norminv(1-0.05/10000,0,1)
ans =
    4.4172
```

Other less stringent multiple comparison corrections such as the false discovery rate (FDR) are also available and implemented in most packages such as SPM, AFNI, FSL, and MATLAB (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001; Genovese et al., 2002).

1.4.3 Random Field Correction

We can obtain a less conservative estimate for (1.11). Under H_0, assuming $\mathbb{E}Y = 0$, we have

$$\begin{aligned}\alpha(h) &= P\Big(\bigcup_{x\in\mathcal{M}} \{Y(x) > h\}\Big)\\ &= 1 - P\Big(\bigcap_{x\in\mathcal{M}} \{Y(x) \le h\}\Big)\\ &= 1 - P\Big(\sup_{x\in\mathcal{M}} Y(x) \le h\Big)\\ &= P\Big(\sup_{x\in\mathcal{M}} Y(x) > h\Big). \qquad (1.14)\end{aligned}$$

In order to construct the α-level test corresponding to H_0, we need to know the distribution of the supremum of field Y. The corresponding p-value based on the supremum of the field, i.e., $\sup_{x\in\mathcal{M}} Y$, is called the *corrected p-value* to distinguish it from the usual p-value obtained from the statistic Y. Note that the p-value is the smallest α-level at which the null hypothesis H_0 is rejected.

Analytically computing the exact distribution of the supremum of random fields is hard in general. The distribution of supremum of Brownian motion is somewhat simple due to its independent increment properties. However, for smooth random fields, it is not so straightforward (Adler, 2000). In Keith Worsley's random field theory, the supremum distribution is based on the expected Euler characteristic (EC). The Euler characteristic approach reformulates the geometric problem as a topological problem (Adler, 1981; Cao and Worsley, 2001; Worsley, 2003; Taylor and Worsley, 2007). For sufficiently high threshold h, it is known that

$$P\Big(\sup_{x\in\mathcal{M}} Y(x) > h\Big) \approx \mathbb{E}\chi(A_h), \tag{1.15}$$

where $A_h = \{x \in \mathcal{M} : Y(x) > h\}$ is the excursion set above x. This indirectly links the problem of statistical inference to that of topology (Figure 1.4).

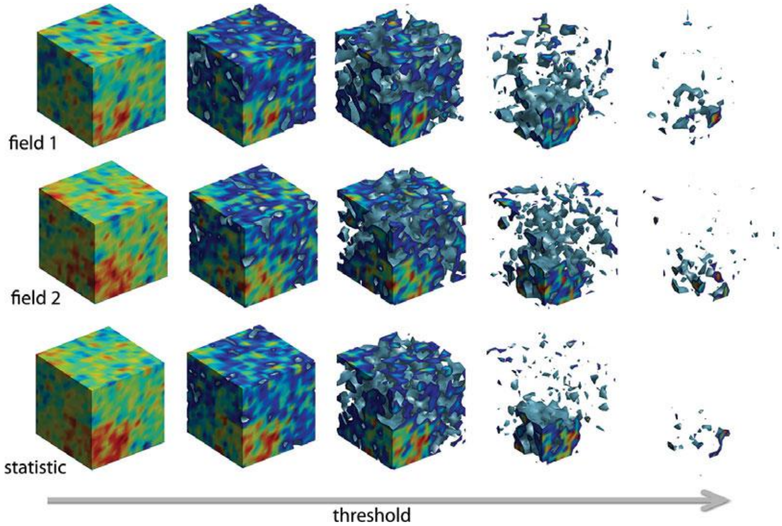


Figure 1.4 Given fields $f_i = \mu + \epsilon_i$, we are interested in detecting the regions of significant signal $\mu > 0$. In the random field theory (Worsley, 2003), we construct a test statistic T out of the fields f_i and determine the topological change of the excursion set $A_h = \{x \in \mathcal{M} : T(x) > h\}$ as we increase the threshold h. This determines the type-I error associated with the hypothesis testing. On the other hand, we can determine the topological change of the individual excursion sets $B_{i,h} = \{x \in \mathcal{M} : f_i(x) > h\}$ first. Then we construct a statistical test on the topological change of $B_{i,h}$ through persistent homology (Chung et al., 2009a).

Compared to other approximation methods such as the Poisson clump heuristic and the tube formulae, the advantage of using the Euler characteristic formulation is that the exact expression can be found for $\mathbb{E}\ \chi(A_h)$ (Worsley et al., 1998; Schmidt and Spodarev, 2005).

There has been a parallel development that tried to link topology to statistical analysis via persistence homology (Bubenik and Kim, 2007; Chung et al., 2009a,b). The use of the mean signal is one way of performing data reduction; however, this may not necessarily be the best way to characterize complex multivariate imaging data. Instead of using the mean signal, we can use topological features such as persistent homology, which pairs local critical values (Edelsbrunner et al., 2002; Zomorodian and Carlsson, 2005; Edelsbrunner and Harer, 2008). It is intuitive that local critical values of $\widehat{\mu}$ approximately characterizes the shape of the continuous signal μ using only a finite number of scalar values. By pairing these local critical values in a nonlinear fashion and plotting them, one constructs the persistence diagram (Cohen-Steiner et al., 2007; Edelsbrunner and Harer, 2008; Morozov, 2008; Zomorodian, 2001). Persistent homology and its application to network modeling will be discussed in Chapter 7.

1.4.4 Type-II Error

Statistical power is an often used measure for determining the necessary sample sizes to achieve a certain statistical significance. The statistical power computation for dependent data or random fields is fairly complicated due to multiple comparisons. The power computation relies on type-II error, which is somewhat involving under the multiple comparisons setting (Hayasaka et al., 2007). Given the null hypothesis H_0 and the alternate hypothesis H_1, type-I and -II errors are defined as follows.

Definition 1.8 *The probabilities of type-II error, denoted as β, is defined as*

$$\begin{aligned}\beta &= P(\textit{Type II error})\\ &= P(\textit{not reject } H_0 \mid H_0\,\textit{false})\\ &= 1 - P(\textit{reject } H_0 \mid H_1 \textit{ true}).\end{aligned}$$

Definition 1.9 *The* power of *the test is defined as* $1 - \beta$.

$$\textit{Power} \ = P(\textit{reject } H_0 \mid H_1 \textit{ ture}).$$

The power of a statistical test is given as the probability of rejecting the null hypothesis that there is no signal when there is an actual signal. When the test procedure has the power of 0.9, it implies that we can correctly reject the null hypothesis 90% of the time when the alternate hypothesis is true. The sample size computation is then based on the power. The power is usually given as a function of sample size and α level. In designing an experiment or collecting data, we are interested in the minimum number of samples in achieving a specific power that is usually set at 85 to 90%.

1.4.5 Statistical Power for T-Test

Let us start with the power computation for scalar measurement at an edge. Consider two samples

$$X_1, \cdots, X_{n_1} \sim N(\mu_1, \sigma^2),$$

$$Y_1, \cdots, Y_{n_2} \sim N(\mu_2, \sigma^2).$$

We are interested in testing

$$H_0 : \mu_1 - \mu_2 = 0 \text{ vs. } H_1 : \mu_1 - \mu_2 = c\sigma \neq 0.$$

The constant c represents the *effect size*, which measures the mean difference with respect to the standard deviation. The effect size c is usually estimated from the sample mean and the standard deviation. For a test statistic, we use the t-statistic with the equal variance assumption:

$$T = \frac{\bar{X} - \bar{Y}}{S_p\sqrt{1/n_1 + 1/n_2}},$$

where $\bar{X}$ and $\bar{Y}$ are the sample means and S_p^2 is the pooled sample variance. If the sample variance of the ith group is denoted by S_i^2, the pooled sample variance is given by

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}.$$

For computing the power, the α-level of the test has to be specified first. Under H_0, the test statistic T follows

$$T \sim t_{n_1+n_2-2},$$

the student t-distribution with $n_1 + n_2 - 2$ degrees of freedom. The rejection region corresponding to the α-level is given by

$$\frac{|\bar{X} - \bar{Y}|}{S_p\sqrt{1/n_1 + 1/n_2}} > t^*_{\alpha/2},$$

where $t^*_{\alpha/2}$ is the quantile satisfying

$$P(T \geq t^*_{\alpha/2}) = \alpha/2.$$

Under H_1, $X_i \sim N(\mu_1, \sigma^2)$ and $Y_i \sim N(\mu_1 - c\sigma, \sigma^2)$. Subsequently, under H_1

$$T' = \frac{\bar{X} - \bar{Y} - c\sigma}{S_p\sqrt{1/n_1 + 1/n_2}} \sim t_{n_1+n_2-2}.$$

Then the power is given by

$$\begin{aligned} \text{Power} &= 1 - P(\text{ none rejection region } |H_1) \\ &= 1 - P\Big(-t^*_{\alpha/2} < T < t^*_{\alpha/2}\Big|H_1\Big) \\ &= 1 - P\Big(-t^*_{\alpha/2} - \frac{c}{\sqrt{1/n_1 + 1/n_2}} < T' < t^*_{\alpha/2} - \frac{c}{\sqrt{1/n_1 + 1/n_2}}\Big). \end{aligned}$$

We approximated σ with S_p.

For sufficiently large n_1 and n_2, we may assume $T \sim N(0,1)$. Let Φ be the cumulative distribution for the standard normal distribution. Note that

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt.$$

In MATLAB, Φ is implemented using `normcdf`. In other nonstatisical computing environments, the error function is more often used. The error function `erf` is defined as

$$\texttt{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2/2}\, dt.$$

The relationship between `normcdf` and `erf` is

$$\texttt{normcdf}(x) = \frac{1}{2} + \frac{1}{2}\texttt{erf}(x/\sqrt{2}).$$

Then using Φ, the power is approximated as

$$\text{Power}(n_1, n_2) = 1 + \Phi\Big(-z^*_{\alpha/2} - \frac{c}{\sqrt{1/n_1 + 1/n_2}}\Big) - \Phi\Big(z^*_{\alpha/2} - \frac{c}{\sqrt{1/n_1 + 1/n_2}}\Big),$$

where $z^*_{\alpha/2}$ is the quantile corresponding to

$$\Phi(z^*_{\alpha/2}) = 1 - \frac{\alpha}{2}.$$

Figure 1.5 displays the power plot for various sample size and the effect size. In general, we should design a method that achieves at least 85% power.

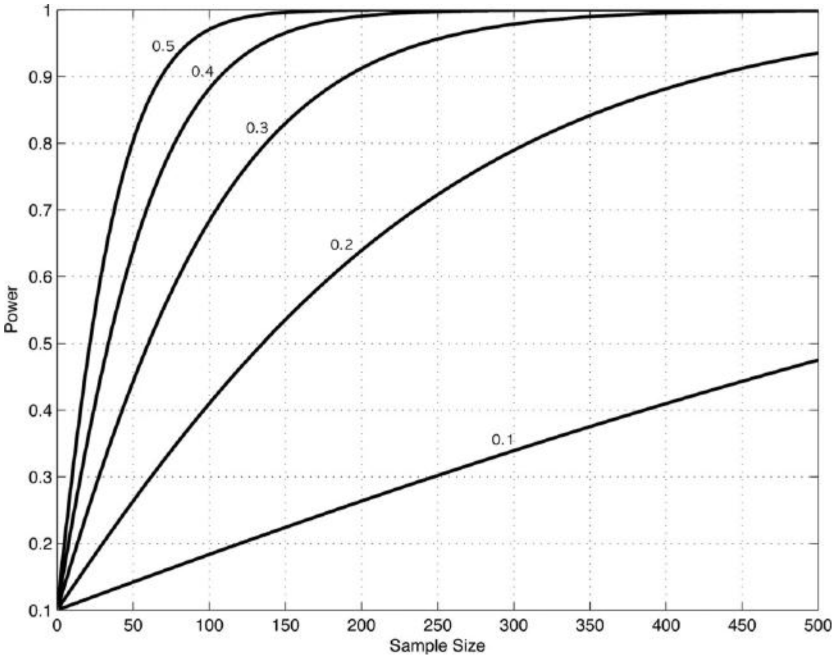


Figure 1.5 Statistical power plot for the two-sample t-test with the effect sizes between 0.1 and 0.5. For the effect size of 0.5, we need at least 50 samples in each group to achieve 80% power.

1.4.6 Power under Multiple Comparisons

In the previous section, the power computation was done at a fixed edge. Assume we have two functional measurements

$$\begin{aligned} X_1(t), \cdots, X_{n_1}(t) &\sim N(\mu_1, \sigma_1^2), \\ Y_1(t), \cdots, Y_{n_2}(t) &\sim N(\mu_2, \sigma_2^2) \end{aligned}$$

over an edge indexed by $t \in \mathcal{M}$ over the whole brain network. X_i and X_j are treated as random fields over $\mathcal{M}$. The usual pointwise hypotheses at each edge $t \in \mathcal{M}$ are given by

$$\begin{aligned} H_0(t) &: \; \mu_1(t) - \mu_2(t) = 0 \\ &vs. \\ H_1(t) &: \; \mu_1(t) - \mu_2(t) = c\sigma > 0. \end{aligned}$$

Instead of the edge-level inference, we need a global inference for the whole network $\mathcal{M}$. The usual global hypotheses accounting for multiple comparisons are then given by

$$\begin{aligned}
&J_0: \ \mu_1(t) - \mu_2(t) = 0 \quad \text{for all} \quad t \in \mathcal{M} \\
&vs. \\
&J_1: \ \mu_1(t) - \mu_2(t) = c\sigma > 0 \quad \text{for some} \quad t \in \mathcal{M}.
\end{aligned}$$

The relationship between the edge-level hypotheses $H_0(t), H_1(t)$ and the global hypotheses J_0, J_1 are

$$J_0 = \bigcap_{t \in \mathcal{M}} H_0(t), \ J_1 = \bigcup_{t \in \mathcal{M}} H_1(t).$$

In order to compute the power over $\mathcal{M}$, it is necessary to determine the type-I error first. Let

$$T(t) = \frac{\bar{X}(t) - \bar{Y}(t)}{S_p(t)\sqrt{1/n_1 + 1/n_2}}$$

be the test random field. Under J_0, $T(t)$, this is a t-random field with $n_1 + n - 2$ degrees of freedom over the whole network $\mathcal{M}$ (Adler, 1981).

We reject J_0 if $T(t) > h$ for some thresholding h for all $t \in \mathcal{M}$. This is equivalent to $\sup_{t \in \mathcal{M}} T(t) > h$. Hence, the type-I error computation requires knowing the distribution of the random variable $\sup_{t \in \mathcal{M}} T(t)$, which can be very involving:

$$\alpha = P\Big(\sup_{t \in \mathcal{M}} T(t) > t_\alpha^*\Big),$$

where t_α^* is the quantile corresponding to the random variable $\sup_{t \in \mathcal{M}} T(t)$. If necessary, the quantile can be determined numerically using the resampling technique (Chung et al., 2014). The rejection region of J_0 corresponding to the α level is given by

$$\mathcal{M}_1 = \Big\{t \in \mathcal{M} : \sup_{t \in \mathcal{M}} T(t) > t_\alpha^*\Big\}.$$

Under J_1, we have

$$X_i(t) \sim N(\mu_1, \sigma^2) \text{ and } Y_i(t) \sim N(\mu_1 + c\sigma, \sigma^2)$$

at $t \in \mathcal{M}_1$. In $\mathcal{M}_0 = \mathcal{M}/\mathcal{M}_1$, we do not reject J_0 and have

$$X_i(t), Y_i(t) \sim N(\mu_1, \sigma^2).$$

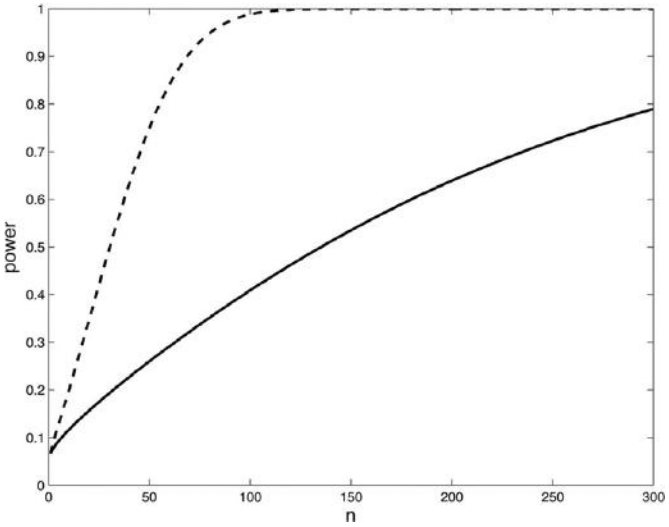


Figure 1.6 Power vs. sample size for the two sample t-test. Solid line is for each voxel and the dotted line is for whole brain surface accounting under multiple comparisons. To obtain power of 0.8 at significance $\alpha = 0.05$, in differentiating the group difference 0.2σ, we need a significantly smaller sample size under multiple comparisons. This is the reason why we usually need smaller sample sizes in imaging studies.

In $\mathcal{M}_1$, we have

$$\begin{aligned} T'(t) &= T(t) - \frac{c\sigma}{S_p(t)\sqrt{1/n_1 + 1/n_2}} \\ &= T(t) - \frac{c}{\sqrt{1/n_1 + 1/n_2}} \sim t_{n_1+n_2-2} \end{aligned}$$

pointwisely. Under the assumption of equal variance field, we have $S_p(t) = \sigma$.

Note that T' is not a t-field in $\mathcal{M}/\mathcal{M}_1$ but a noncentral t-field (Hayasaka et al., 2007). Then the overall power over $\mathcal{M}$ is given as

$$\begin{aligned} \text{Power } &= P\Big(\sup_{t\in\mathcal{M}} T(t) > t_\alpha^* \Big| J_1\Big) \\ &= P\Big(\sup_{t\in\mathcal{M}_1} T'(t) > t_\alpha^* - \frac{c}{\sqrt{1/n_1 + 1/n_2}}\Big). \end{aligned}$$

(See Figure 1.6.)

2
Brain Network Nodes and Edges

The human brain consists of many functionally and structurally distinct areas, each receiving and sending various neuronal projections from and to each other (Young, 1992). The brain forms a very complex network of connections (Young, 1992). Neuroanatomical connectivity places structural constraints on the functional connectivity of the cerebral cortex (Sporns et al., 2000). It is crucial to understand functional connectivity in relation to the underlying structural connectivity. In the usual brain connectivity studies, we usually have to either manually or automatically define regions of these distinct areas that can serve as network nodes. In this chapter, we review various technique for identifying network nodes and constructing network edges using structural brain images.

2.1 Brain Templates

Brain network nodes are often defined through existing template and manual parcellations on top of the templates. In the past, the Talairach atlas was used as the standard template. The Talairach atlas has Brodmann's areas labeled in an approximate fashion (Talairach and Tournoux, 1988). Due to many limitations, the Montreal Neurological Institute (MNI) defined a new standard brain by using a large cross-sectional magnetic resonance images (MRI) of adult normal subjects. Brain processing and analysis packages such as Statistical Parametric Mapping (SPM), Analysis of Functional Neuroimages (AFNI), and FMRIB Software Library (FSL) all use the standard brains from the MNI. However, over the years, there are many variations and improvements in the MNI template. So care should be taken to understand what the specific MNI template is in the software packages and studies.

The current standard MNI template is the *ICBM152*, which is the average of 152 MRI scans of normal subjects. The International Consortium for Brain Mapping adopted this as its standard template. Colin Holmes, an MNI researcher, was scanned 27 times, and the scans were coregistered and averaged to create a detailed single-subject MRI. This average was also matched to the MNI305, to create the image known as *colin27*.

2.2 Brain Parcellations

The whole brain is often parcellated into p disjoint regions, where p is between 50 to 200 (Hagmann et al., 2007; Gong et al., 2009; Fornito et al., 2010; Zalesky et al., 2010). Harvard–Oxford atlas is a probabilistic parcellation covering 48 cortical and 21 subcortical areas (Desikan et al., 2006). T1-weighted images of 21 male and 16 female subjects (ages 18 to 50) were individually segmented semiautomatically. The T1-weighted images were affine-registered to MNI152 space using FMRIB's linear image registration tool (FLIRT) in FSL, and the individual labels were aligned using the same affine transformation. The transformed labels were then combined across subjects to form a population probability map for each label. LONI (Laboratory of Neuro Imaging) probabilistic brain atlas (LPBA40) is constructed similar to Harvard–Oxford atlas but using different processing pipelines and using 40 subjects (Shattuck et al., 2008).

Although most existing parcellations are based on the parcellation of gray matter, diffusion-tensor based white matter parcellations such as ICBM-DTI-81 are also available (Mori et al., 2008).

2.2.1 Anatomical Automatic Labeling (AAL)

Anatomical Automatic Labeling (AAL) is probably the most often used parcellation scheme in automatically identifying multiple regions of interest (ROI) in the brain. AAL parcellation provides 116 labels for all the cortical and subcortical structures in the MNI template space (Figures 2.1 and 5.4) (Tzourio-Mazoyer et al., 2002). AAL parcellation in MATLAB format is also available.[1] Inside the package, the structured array `ROI.ID` contains the integer label for parcellations. The first five integer labels are 2001, 2002, 2101, 2102, and 2111. The structured array `ROI.Nom_L.mat` contains the

[1] http://neuro.imm.dtu.dk/wiki/Automated_Anatomical_Labeling

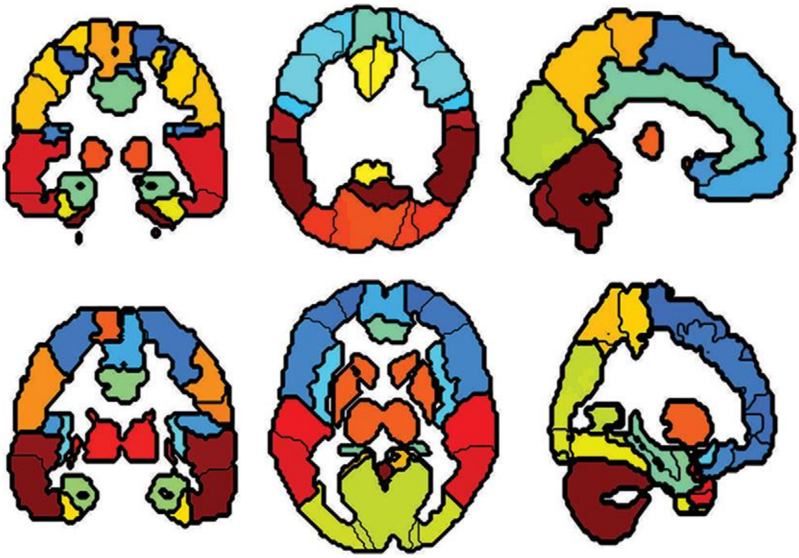


Figure 2.1 AAL parcellation with 116 disjoint regions. Each region is colored differently. AAL mainly parcellate gray matter regions.

corresponding anatomical labels for 116 regions as strings. The first five labels are the following:

```
'Precentral_L'
'Precentral_R'
'Frontal_Sup_L'
'Frontal_Sup_R'
'Frontal_Sup_Orb_L'
```

where `Precentral_L` and `Precentral_R` are the left and right precentral gyri. For convenience, a short-hand notations are available in `ROI.Nom_C`. The first five short-hand notations for the aforementioned regions are `'FAG'`, `'FAD'`, `'F1G'`, `'F1D'`, `'F1OG'`. AAL labels are often used in reporting findings in brain network analysis. AAL labels are often used to report the node-level findings in neuroimaging literature (Chung et al., 2017b). The complete list of ROI labels and names are given in Table 2.1.[2]

One may superimpose functional activations such as fMRI and PET on top of parcellation ROI and can set up a model of how activation in one region is related to other regions. For instance, we can set up linear models of how

[2] Table was complied by Andrey Gritsenko of University of Wisconsin–Madison.

Table 2.1. *AAL parcellation short-hand notation and labels.*

Index	Notation	Brain region
1	FAG	Left precentral gyrus
2	FAD	Right precentral gyrus
3	F1G	Left superior frontal gyrus, dorsolateral
4	F1D	Right superior frontal gyrus, dorsolateral
5	F1OG	Left superior frontal gyrus, orbital part
6	F1OD	Right superior frontal gyrus, orbital part
7	F2G	Left middle frontal gyrus, lateral part
8	F2D	Right middle frontal gyrus, lateral part
9	F2OG	Left middle frontal gyrus, orbital part
10	F2OD	Right middle frontal gyrus, orbital part
11	F3OPG	Left opercular part of inferior frontal gyrus
12	F3OPD	Right opercular part of inferior frontal gyrus
13	F3TG	Left area triangularis
14	F3TD	Right area triangularis
15	F3OG	Left orbital part of inferior frontal gyrus
16	F3OD	Right orbital part of inferior frontal gyrus
17	ORG	Left rolandic operculum
18	ORD	Right rolandic operculum
19	SMAG	Left supplementary motor area
20	SMAD	Right supplementary motor area
21	COBG	Left olfactory cortex
22	COBD	Right olfactory cortex
23	FMG	Left superior frontal gyrus, medial part
24	FMD	Right superior frontal gyrus, medial part
25	FMOG	Left superior frontal gyrus, medial orbital part
26	FMOD	Right superior frontal gyrus, medial orbital part
27	GRG	Left gyrus rectus
28	GRD	Right gyrus rectus
29	ING	Left insula
30	IND	Right insula
31	CIAG	Left anterior cingulate gyrus
32	CIAD	Right anterior cingulate gyrus
33	CINMG	Left middle cingulate
34	CINMD	Right middle cingulate
35	CIPG	Left posterior cingulate gyrus
36	CIPD	Right posterior cingulate gyrus
37	HIPPOG	Left hippocampus
38	HIPPOD	Right hippocampus
39	PARA_HIPPOG	Left parahippocampal gyrus
40	PARA_HIPPOD	Right parahippocampal gyrus
41	AMYGDG	Left amygdala
42	AMYGDD	Right amygdala
43	V1G	Left calcarine sulcus
44	V1D	Right calcarine sulcus
45	QG	Left cuneus
46	QD	Right cuneus
47	LINGG	Left lingual gyrus

Table 2.1. (*cont.*)

48	LINGD	Right lingual gyrus
49	O1G	Left superior occipital
50	O1D	Right superior occipital
51	O2G	Left middle occipital
52	O2D	Right middle occipital
53	O3G	Left inferior occipital
54	O3D	Right inferior occipital
55	FUSIG	Left fusiform gyrus
56	FUSID	Right fusiform gyrus
57	PAG	Left postcentral gyrus
58	PAD	Right postcentral gyrus
59	P1G	Left superior parietal lobule
60	P1D	Right superior parietal lobule
61	P2G	Left inferior parietal lobule
62	P2D	Right inferior parietal lobule
63	GSMG	Left supramarginal gyrus
64	GSMD	Right supramarginal gyrus
65	GAG	Left angular gyrus
66	GAD	Right angular gyrus
67	PQG	Left precuneus
68	PQD	Right precuneus
69	LPCG	Left paracentral lobule
70	LPCD	Right paracentral lobule
71	NCG	Left caudate nucleus
72	NCD	Right caudate nucleus
73	NLG	Left putamen
74	NLD	Right putamen
75	PALLG	Left globus pallidus
76	PALLD	Right globus pallidus
77	THAG	Left thalamus
78	THAD	Right thalamus
79	HESCHLG	Left transverse temporal gyri
80	HESCHLD	Right transverse temporal gyri
81	T1G	Left superior temporal gyrus
82	T1D	Right superior temporal gyrus
83	T1AG	Left superior temporal pole
84	T1AD	Right superior temporal pole
85	T2G	Left middle temporal gyrus
86	T2D	Right middle temporal gyrus
87	T2AG	Left middle temporal pole
88	T2AD	Right middle temporal pole
89	T3G	Left inferior temporal gyrus
90	T3D	Right inferior temporal gyrus
91	CERCRU1G	Left crus I of cerebellar hemisphere
92	CERCRU1D	Right crus I of cerebellar hemisphere
93	CERCRU2G	Left crus II of cerebellar hemisphere
94	CERCRU2D	Right crus II of cerebellar hemisphere
95	CER3G	Left lobule III of cerebellar hemisphere
96	CER3D	Right lobule III of cerebellar hemisphere

Table 2.1. (*cont.*)

97	CER4_5G	Left lobule IV, V of cerebellar hemisphere
98	CER4_5D	Right lobule IV, V of cerebellar hemisphere
99	CER6G	Left lobule VI of cerebellar hemisphere
100	CER6D	Right lobule VI of cerebellar hemisphere
101	CER7BG	Left lobule VIIB of cerebellar hemisphere
102	CER7BD	Right lobule VIIB of cerebellar hemisphere
103	CER8G	Left lobule VIII of cerebellar hemisphere
104	CER8D	Right lobule VIII of cerebellar hemisphere
105	CER9G	Left lobule IX of cerebellar hemisphere
106	CER9D	Right lobule IX of cerebellar hemisphere
107	CER10G	Left lobule X of cerebellar hemisphere (flocculus)
108	CER10D	Right lobule X of cerebellar hemisphere (flocculus)
109	VER1_2	Lobule I, II of vermis
110	VER3	Lobule III of vermis
111	VER4_5	Lobule IV, V of vermis
112	VER6	Lobule VI of vermis
113	VER7	Lobule VII of vermis
114	VER8	Lobule VIII of vermis
115	VER9	Lobule IX of vermis
116	VER10	Lobule X of vermis (nodulus)

fMRI in an ROI is dependent on all other ROI (Lee et al., 2012; Chung et al., 2015a). From the linear models, we can further obtain correlations that can be used to build a connectivity matrix that characterizes the whole brain network. Unlike other static imaging modalities such as FDG-PET and diffusion tensor imaging (DTI), fMRI has the temporal component so it is possible to code the cause and effect as directional information in the connectivity matrix as well.

The major shortcoming of using the existing parcellations including AAL is the lack of refined spatial resolution. Even if we detected connectivity differences between large chunks of brain regions, it is not possible to localize what parts of parcellations are affected without additional analysis. There is a strong need to develop a higher-resolution parcellation scheme.

It is possible to subdivide existing parcellations into smaller disjoint subregions. Mostly based on spectral clustering (Craddock et al., 2012; Pepe et al., 2015) and graph cuts (Shen et al., 2010), many algorithms have been proposed for subdividing parcellations mostly in fMRI studies. However, many of these methods do not preserve the hierarchical nestedness. Parcellations at one scale will topologically conflict with parcellations at different scales.

Numerous methods have been proposed for automatically identifying ROI that are often based on clustering algorithms or region growing methods that increase the similarity of functional activation within cluster (Craddock et al., 2012). Among all clustering algorithms, spectral clustering including graph cuts seems to be most popular and effective in producing ROI of uniform size.

2.2.2 Supervoxel Functional Parcellation

It is also possible to parcellate the brain regions using the resting-state (rs-fMRI) (Thirion et al., 2014). For clustering fMRI, we can use the simple linear iterative clustering (SLIC) supervoxel algorithm that adopts k-means clustering but has lower computational complexity $O(k)$ (Achanta et al., 2012; Gritsenko, 2018). The standard k-means clustering searches the entire brain images while SLIC searches a limited region. A search for similar voxels is done in a cubical region around the supervoxel center. Depending on applications, various modifications to the distance used in k-means clustering are possible. This requires computing the similarity measure or distance

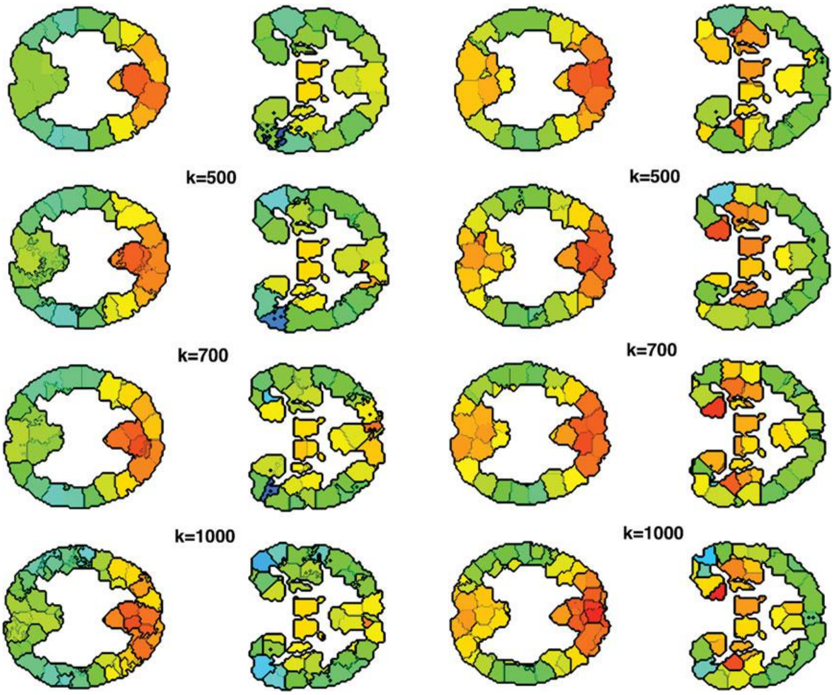


Figure 2.2 Supervoxel segmentation via SLIC with $k = 500$, 700, and 1,000 parcellations on rs-fMRI (Gritsenko, 2018). The figure was generated by Andrey Gritsenko of University of Wisconsin–Madison.

between fMRI in two different voxels, a topic we will discuss in a later chapter. Figure 2.2 shows the results of supervoxel segmentation using a subset of 100 resting-sate fMRI from the human connectome project (HCP) (Van Essen et al., 2012). The correlation between rs-fMRI signals is used in constructing the distance, a topic we will study in a later chapter. Although segmentation obtained using SLIC does not follow AAL parcellation, there are still well-marked correspondences between them.

2.3 Deterministic Connectivity

Structural connectivity studies have been popular in recent years due to the advancement of DTI, which is an MRI technique that has been used to characterize the macrostructure of biological tissues using magnitude, anisotropy, and anisotropic orientation associated with water diffusion in the brain (Basser et al., 1994). DTI provides directional and connectivity information that the standard MRI usually does not provide and estimates the patterns of white matter connectivity. The white matter fibers pose a physical constraint on the movement of water molecules along the direction of fibers. It is assumed that the direction of greatest diffusivity is most likely aligned to the local orientation of the white matter fibers. White matter fibers consist mostly of myelinated axons that connect gray matter regions of the brain to each other. The axons are filled with neuronal filaments running along its longitudinal axis, which contributes to the anisotropy of water diffusion (Mori and van Zijl, 2002).

DTI can be used to characterize the structural connectivity of the human brain noninvasively by tracing white matter fiber tracts. DTI tractography algorithms can trace up to a half million tracts per brain, which serve as edges in network construction. The whole brain has been traditionally parcellated into p disjoint regions, where p is between 50 to 200. White matter fibers provide information of how one region is connected to another via a p-by-p connectivity matrix. The connectivity matrix is then thresholded to produce a binarized adjacency matrix, which is often used in graph theory–based analysis (Hagmann et al., 2007; Fornito et al., 2010; Gong et al., 2009; Zalesky et al., 2010; Chung et al., 2017b).

Structural brain connectivity can be modeled as a network graph using white matter fiber bundles obtained from DTI tractography algorithms. White matter tractography offers the unique opportunity to characterize the trajectories of white matter fiber bundles noninvasively in the brain. Various deterministic tractographies have been used to visualize and map out major white matter pathways in individuals and brain atlases (Conturo et al., 1999; Mori et al., 1999; Basser et al., 2000; Catani et al., 2002; Mori and van Zijl, 2002;

Lazar et al., 2003; Thottakara et al., 2006; Yushkevich et al., 2007); however, tractography data can be challenging to interpret and quantify. Recent efforts have attempted to cluster (O'Donnell et al., 2006) and automatically segment white matter tracts (O'Donnell and Westin, 2007) as well as characterize tract shape parameters (Batchelor et al., 2006). Many of these techniques can be quite computationally demanding. Efficient methods for extracting tracts, representing tract shape, regional tract segmentation and clustering, tract registration, and quantification would be of tremendous value to researchers.

DTI registration. DTI usually goes through image preprocessing before overlaying the fiber tracts to existing parcellations (Hanson et al., 2013; Kim et al., 2015). DTI needs to be corrected for eddy current–related distortion and head motion. FSL software[3] can be often used for the purpose. The distortions from field inhomogeneities need to be corrected before performing a tensor estimation (Jezzard and Clare, 1999; Cook et al., 2006). CAMINO is often used for the nonlinear tensor estimation. Spatial normalization of DTI plays a key role in constructing brain network graphs that are spatially compatible across different subjects. The quality of spatial normalization determines the extent to which white matter tracts are aligned. It has direct impacts on the successful removal of shape confounds and consequently on the validity, specificity, and sensitivity of the subsequent statistical inferences of group differences. Inadequate normalization with coarse registration algorithms can result in insufficient removal of shape differences that is necessary for obtaining topologically invariant network graph. Spatial normalization of DTI data often requires a diffeomorphic registration strategy (Joshi et al., 2004; Zhang et al., 2007a). DTI-ToolKit (DTI-TK)[4] can be used (Zhang et al., 2007b). This approach combines full tensor coregistration and high-dimensional diffeomorphic spatial normalization. The registration is based on an iterative strategy (Joshi et al., 2004; Zhang et al., 2007b) where the initial template is computed as the average of original DTI. Then DTI is first affinely aligned to the template. The tensor images after the affine alignment are then provided as the input to the registration algorithm. The algorithm leverages full tensor-based similarity metrics while optimizing tensor orientation explicitly. The metric is based on the L_2-distance between the anisotropic parts of diffusion profiles associated with the diffusion tensors (Zhang et al., 2006). The algorithm then approximates smooth transformations using a dense piecewise affine parameterization, which is sufficient when the required deformations are not large. Now compute a refined template as an average of the normalized images. If the change between templates from consecutive iterations is sufficiently

[3] www.fmrib.ox.ac.uk/fsl
[4] www.nitrc.org/projects/dtitk

small, we stop the iteration; otherwise we continue the iterative process of getting a new template and refitting.

Streamline tractography. The white matter connectivity is mainly obtained by the streamline based tractography, in which a continuous path of connection between two brain regions is estimated as a streamline whose tangential velocity field is given by the principal eigenvectors. Most of current white matter tractography is based on streamlines (Conturo et al., 1999; Mori et al., 1999; Basser et al., 2000) or its variations such as tensor deflection (TEND) method (Lazar et al., 2003). Whole brain tractography studies routinely generate up to a half million tracts per brain, which serves as edges in an extremely large 3D graph. The directional information of water diffusion is usually represented as a symmetric positive definite 3 × 3 matrix $D = (d_{ij})$ which is usually called as the *diffusion tensor* or *diffusion coefficients*. The diffusion tensors are often normalized by its transpose, i.e., $D/\mathrm{tr}D$. This normalization guarantees that the sum of eigenvalues of D equals 1. The eigenvectors and eigenvalues of D are obtained by solving

$$D\mathbf{v} = \lambda\mathbf{v},$$

which results in three eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and the corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$. We may assume that the eigenvectors are normalized as $\|\mathbf{v}_j\| = 1$. The principal eigenvector $\mathbf{v}_1$, the eigenvector corresponding to the largest eigenvalue, usually determines the direction of the water diffusion, and is mainly used in streamline-based tractography.

For the given principal vector fields $\mathbf{v}_1$, the corresponding streamline $p = \psi(t)$ satisfies the ordinary differential equation

$$\frac{d\psi}{dt} = \mathbf{v}_1 \circ \psi(t) \tag{2.1}$$

This ordinary differential equation gives a family of integral curves whose tangent vector is $\mathbf{v}_1$ (Betounes, 1998). By integrating (2.1) with respect to the parameter t, we obtain the equivalent integral tranform

$$\psi(t) = \int_0^t \mathbf{v}_1 \circ \psi(t)\, dt + \psi(0). \tag{2.2}$$

The most common numerical methods for solving (2.1) and (2.2) are Euler's method and the Runge–Kutta algorithm (Basser et al., 2000).

The streamline-based techniques for obtaining fiber tracts have three main steps: (1) defining seed points, (2) performing integration (2.2), and (3) determining stopping criteria (Vilanova et al., 2004). The stopping criteria avoids the area where the principal vector fields cannot be robustly obtained.

Note that streamlines have been encountered in the context of estimating cortical thickness using the Laplace equation (Jones et al., 2000; Chung, 2012).

2.3.1 Electrical Circuit Model

In this section, we present an electrical circuit model for constructing connectivity matrices. The electrical circuit model can be viewed as a generalization of the simple tract counting method. Let's start with the arithmetic and harmonic means. Given measurements $R_1, \cdots, R_k$, their arithmetic mean $A(R_1, \cdots, R_k)$ is given by the usual sample mean, i.e.,

$$A(R_1, \cdots, R_k) = \frac{1}{k} \sum_{i=1}^{k} R_i.$$

The tract count and its mean are based on the arithmetic addition. The *harmonic mean* $H(R_1, \cdots, R_k)$ of $R_1, \cdots, R_k$ is given by

$$H(R_1, \cdots, R_k) = \frac{k}{\frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_k}}.$$

The harmonic mean is given by the reciprocal of the arithmetic mean of reciprocal of measurements:

$$H(R_1, \cdots, R_k) = 1 \Big/ \Big(\frac{1}{R_1}, \cdots, \frac{1}{R_k} \Big).$$

The harmonic mean has been mainly used in measuring the rates of physical systems such as the speed of a car (Zhang et al., 1999) or resistance of electrical circuits (Chung, 2012). Beyond physical systems, it has been used in k-means clustering (Zhang et al., 1999), where the harmonic k-mean is used instead of the usual arithmetic mean. The harmonic mean has been also used in the integrated likelihood for the Bayesian model selection problem (Raftery et al., 2006). Whenever we deal with rates and ratio-based measures such as resistance, the harmonic mean provides more robust and accurate average compared to the arithmetic mean and often is used in various branch of sciences. The use of harmonic means can naturally incorporate the length of tracts in connectivity.

Motivated by Doyle and Snell (1984), the structural brain network can be analogously modeled as an electrical system consisting of series and parallel circuits (Figure 2.3). Each fiber tract may be viewed as a single wire with resistance R proportional to the length of the wire. If two regions are connected

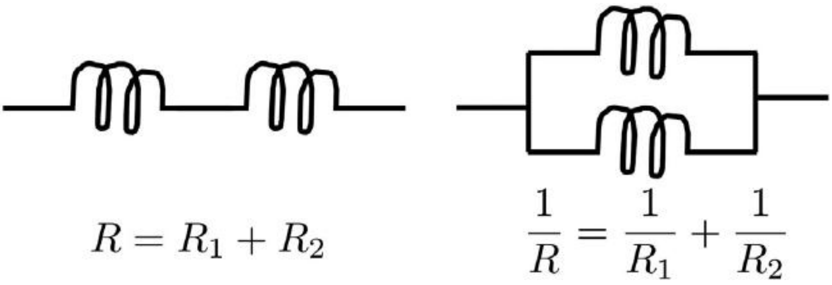


Figure 2.3 Examples of series circuit (left) and parallel circuit (right).

through an intermediate region, it forms a series circuit. In the series circuit, the total resistance R is additive so we have

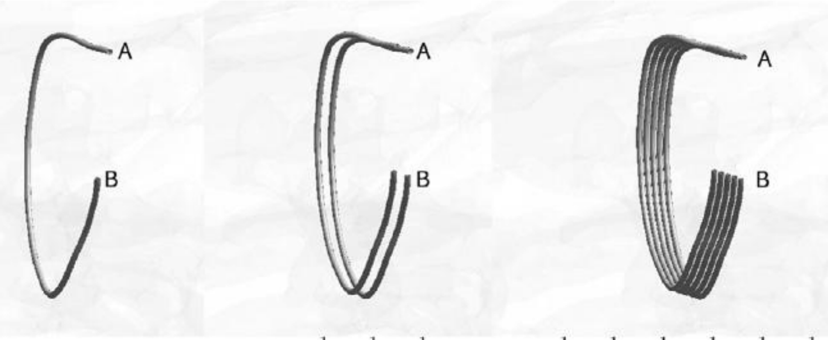
$$R = R_1 + \cdots + R_k,$$

where R_k is the resistance of the kth tract. If multiple fiber tracts connect two regions, it forms a parallel circuit, where the total resistance is

$$\frac{1}{R} = \frac{1}{R_1} + \cdots + \frac{1}{R_k}. \tag{2.3}$$

The total resistance of a series circuit is related to the arithmetic mean of tract lengths. On the other hand, the total resistance (2.3) of a parallel circuit is related to the harmonic mean. Any complex parallel circuits in an electrical system can be simplified using a single wire with the equivalent resistance. Hence, we can simplify whole brain fiber tracts into a smaller number of equivalent tracts in the model. The reciprocal of the resistance is then taken as the measure of connectivity. Smaller resistance corresponds to stronger connectivity. Figure 2.4 shows examples of parallel circuits. If all the tracts are 10 cm in length, the total resistance becomes 10, 5, and 2 as the number of tracts increases to 1, 2, and 5. The corresponding connectivities between A and B are 0.1, 0.2, and 0.5. Thus, if the tract lengths are all the same, the resistance-based connectivity is proportional to tract counts. Figure 2.5 shows various toy networks and the corresponding resistance matrices. The corresponding resistance matrices are as follows:

$$\begin{pmatrix} 0 & 1 & 1 & \infty \\ 1 & 0 & \infty & \infty \\ 1 & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{pmatrix} \begin{pmatrix} 0 & 1/2 & 1/2 & \infty \\ 1/2 & 0 & \infty & \infty \\ 1/2 & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{pmatrix}$$



$R = 10$

$\frac{1}{R} = \frac{1}{10} + \frac{1}{10}$

$R = 5$

$\frac{1}{R} = \frac{1}{10} + \frac{1}{10} + \frac{1}{10} + \frac{1}{10} + \frac{1}{10}$

$R = 2$

Figure 2.4 Multiple fiber tracts connecting the regions A and B are modeled as a parallel circuit. The resistance in a wire is proportional to the length of the wire. As more tracts connect the regions in parallel, the resistance decreases and the strength of connection increases. In this example, we let the resistance of each tract equal to the length of the tract. If all the tracts are 10 cm in length, the total resistance becomes 10, 5, and 2 as the number of tracts increases to 1, 2, and 5. The connectivity between A and B is defined as the reciprocal of resistance. The corresponding connectivities are 0.1, 0.2, and 0.5.

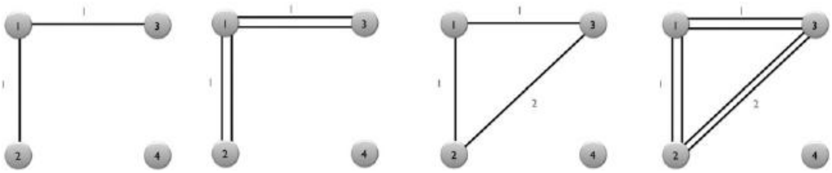


Figure 2.5 Toy networks with different connectivity resistance. The numbers between nodes are the length of tracts.

$$\begin{pmatrix} 0 & 1 & 1 & \infty \\ 1 & 0 & 2 & \infty \\ 1 & 2 & 0 & \infty \\ \infty & \infty & \infty & 0 \end{pmatrix} \begin{pmatrix} 0 & 1/2 & 1/2 & \infty \\ 1/2 & 0 & 1 & \infty \\ 1/2 & 1 & 0 & \infty \\ \infty & \infty & \infty & 0 \end{pmatrix}.$$

The resistance between indirectly connected nodes is ∞. The most redundant network has the smallest resistance. The reciprocal of the resistance is taken as the strength of connectivity.

The electronic circuit model can be used in constructing a simplified but equivalent brain network. The two end points of tracts are identified. All the parallel tracts between any two regions are identified and replaced with a single tract with the equivalent resistance. This process completely removes all the parallel circuits. At the end, the simplified circuit forms a graph with the resistances as the edge weights.

Consider a tract $\mathcal{M}$ consisting of n control points $p_1, \cdots, p_n$ obtained through tractography algorithms. Consider an inverse map ζ^{-1} that maps the control point p_j onto the unit interval as

$$\zeta^{-1} : p_j \rightarrow \frac{\sum_{i=1}^{j} \|p_i - p_{i-1}\|}{\sum_{i=1}^{n} \|p_i - p_{i-1}\|} = t_j \ (2 \leq j \leq n), \tag{2.4}$$

where $\|\cdot\|$ is the Euclidean distance. This is the ratio of the arc length from the point p_1 to p_j, to p_1 to p_n. We let this ratio be t_j. We assume $\zeta^{-1}(p_1) = 0$. The ordering of the control points is also required in obtaining smooth one-to-one mapping. We parameterize the tract on a unit interval:

$$\zeta : [0,1] \rightarrow \mathcal{M}.$$

Then the total length $L(\mathcal{M})$ of the tract $\mathcal{M}$ is given by

$$L(\mathcal{M}) = \int_0^1 d\zeta(t)$$

and discretely approximated as

$$L(\mathcal{M}) = \sum_{i=1}^{n-1} \|\zeta(t_{i+1}) - \zeta(t_i)\| = \sum_{i=1}^{n-1} \|p_{i+1} - p_i\|.$$

The connectivity matrices are subsequently determined by computing the resistance between parcellations (Figure 2.6-middle). We took the reciprocal of the resistance between the nodes as entries of the connectivity matrix. Smaller resistance corresponds to stronger connection. The method replaces a collection of parallel circuits with a single equivalent circuit. This process completely removes all the parallel circuits and simplifies complex parallel circuits to a simple circuit. The simplified circuit naturally forms a 3D graph with the resistances as the edge weights. Figure 2.7-middle shows the mean connectivities using the graph representation.

Fractional anisotropy (FA) can be also used in constructing the connectivity matrix. FA may provide additional structural information that the tractography alone may not provide. Note that most parcellations are located in the gray matter regions, where the fiber tracts starts and ends, so FA-values are expected to be very low at the nodes. The mass center of each parcellation is taken as a node of the network. The mean FA value at the node positions is 0.18 ± 0.10. In fact, at each node position, the two-sample t-statistic did not yield any group

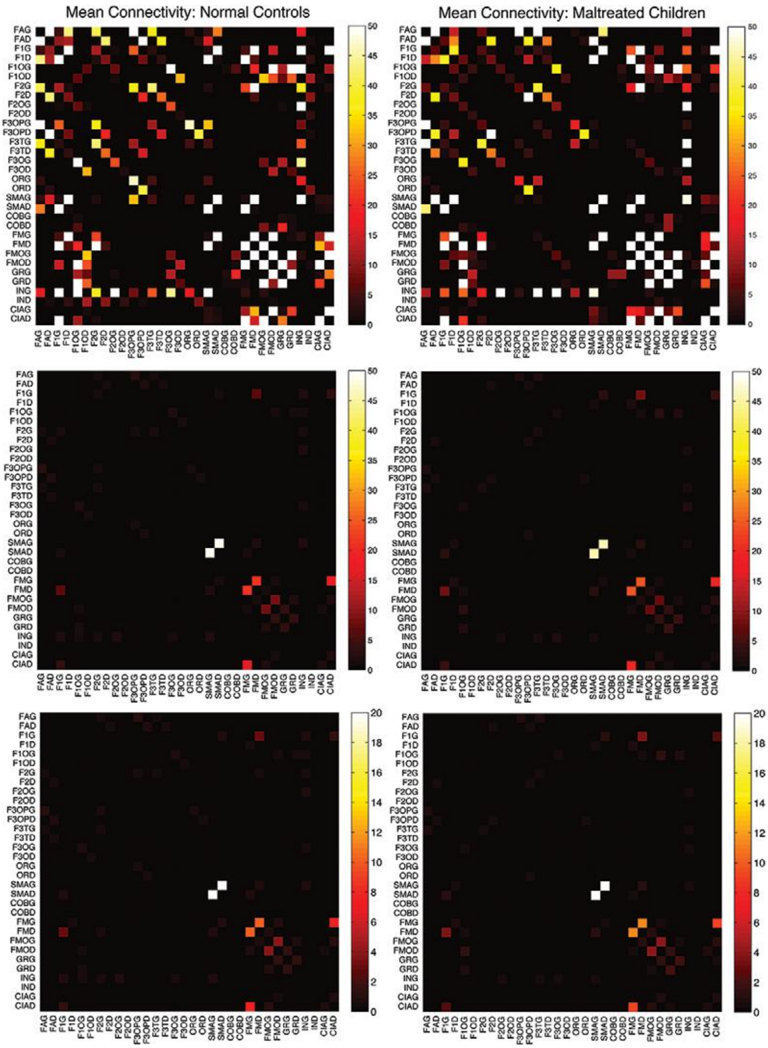


Figure 2.6 Top: mean connectivity matrices for first 32 nodes for normal controls (left) and maltreated children (right) using tract counts (Chung et al., 2017b). Middle: the electrical resistance based connectivity matrices. Bottom: resistance-based mean connectivity matrices that incorporate FA-values.

differences at 0.05 significance. However, along the tracts, it is expected FA-values are higher and this may influence the connectivity. Figure 2.8 shows an example of how FA-values change along 108 tracts between the left superior motor area (SMAG) to the right superior motor area (SMAD).

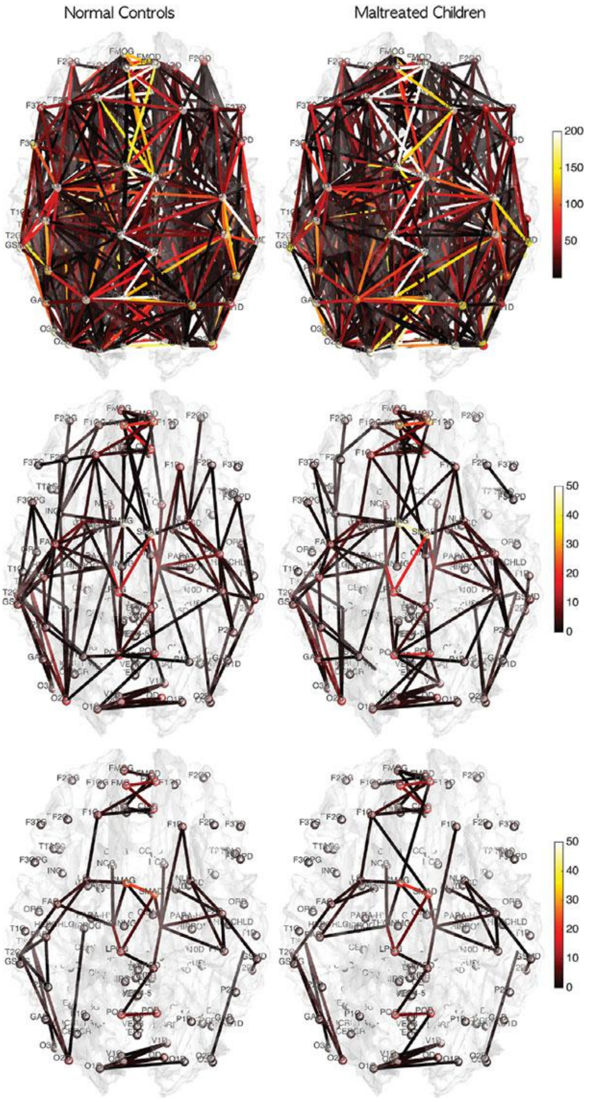


Figure 2.7 Mean connectivity of normal controls and maltreated children (Chung et al., 2017b). The color of nodes and edges correspond tract counts (top), electrical resistance (middle), and electrical resistance with FA-values (bottom). Strong connections are consistently shown in all three models.

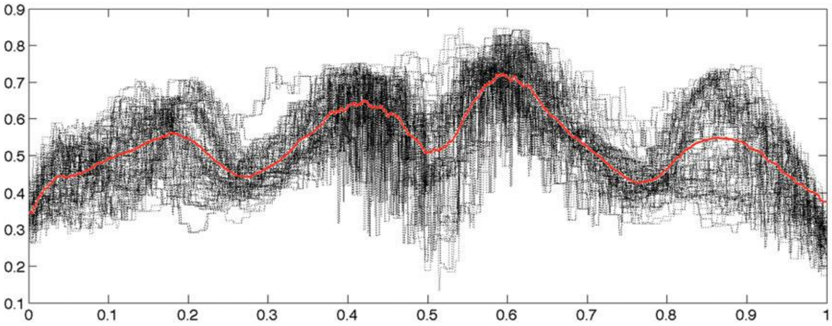


Figure 2.8 FA-values along 108 tracts between the left SMAG to the right SMAD displayed for a subject. The tracts are reparameterized between 0 and 1 from the left to right hemisphere. The red line is the average of FA-values of all the tracts.

We incorporate FA-values along the fiber tracts as follows. By identifying voxels that tracts are passing through, we can linearly interpolate the FA-values along the tracts. If the FA value is larger at a certain part of a tract, it is more likely that the segment of the tract has been more stably estimated. Thus, the resistance of a tract segment can be modeled as inversely proportional to the FA value but proportional to the length of the segment $d\zeta$ at point:

$$dR \propto \frac{d\zeta}{\text{FA}}.$$

Note that we are using the reciprocal of the resistance as a tract connectivity metric C, i.e.,

$$C = \frac{1}{dR} = \frac{\text{FA}}{d\zeta}.$$

So heuristically, the larger the value of FA, stronger the connectivity in the tract segment. Subsequently, the total resistance R of the whole whole tract is defined as

$$R = \int_0^1 \frac{1}{\text{FA}(\zeta(t))}\, d\zeta(t). \tag{2.5}$$

If the tractography processing is properly done, it is not possible to have zero FA-values along the obtained tracts so the integral (2.5) is well defined. The integral (2.5) is discretized as

$$R = \sum_{i=1}^{n-1} \frac{\|\zeta(t_{i+1}) - \zeta(t_i)\|}{\text{FA}(\zeta(t_i))}.$$

We may fix the step size $\|\zeta(t_{i+1}) - \zeta(t_i)\|$ at 0.1 mm. Thus,

$$R = \sum_{i=1}^{n-1} \frac{0.1}{\text{FA}(\zeta(t_i))}.$$

This is the weighted version of the resistance that incorporates FA-values. The connectivity matrices are then similarly determined by computing the reciprocal of the weighted resistance between parcellations (Figure 2.6-bottom). Smaller resistance corresponds to stronger connection. The pattern of connectivity matrices is almost identical to the connectivity without FA-values although the scale and local variations differ slightly. This indicates the robustness of the resistance-based connectivity metric.

2.3.2 Tract Counts vs. Tract Length–Based Connectivity

In the previous section, we presented various methods for constructing structural connectivity matrices. However, it is unclear how they are related and if they will give consistent statistical results at the end. Suppose there are k tracts between two parcellations. Suppose the tract lengths are all identical as L. Then the tract count-based connectivity gives the connectivity strength k. The length-based connectivity gives the connectivity strength k/L. Under the ideal situation of the same tract length, the tract length-based connectivity is proportional to tract count-based connectivity. For the incorporation of FA-values to the connectivity, if we assume the identical FA-values along the tract, we have FA $\cdot\, k/L$ as connectivity. All three methods are proportional to each other. Thus, in the ideal situation of the same tract length and FA-values, the final statistical analyses for three methods would be identical since most statistical test procedures are scale invariant.

The difference arises in practice where the tract lengths are all different. This is a difficult problem we can't address theoretically. To begin to deal with this issue, we checked the robustness of the tract length-based method in relation to the tract count method. To determine the robustness of the tract length-based metric, we calculated the relative error in comparison to traditional tract count-based connectivity in the tract mislabeling problem.

Figure 2.9 shows a schematic of a possible tract mislabeling problem. Assume k number of tracts are passing through between parcellations 2 and 3 (top). In the traditional connectivity metric, tract count is used as the strength of connectivity. So the expected connectivity is $C_E = k$. However, m number of tracts might be mislabeled to pass through parcellations 1 and 3 (bottom). So the observed connectivity is given by $C_O = k - m$. The relative

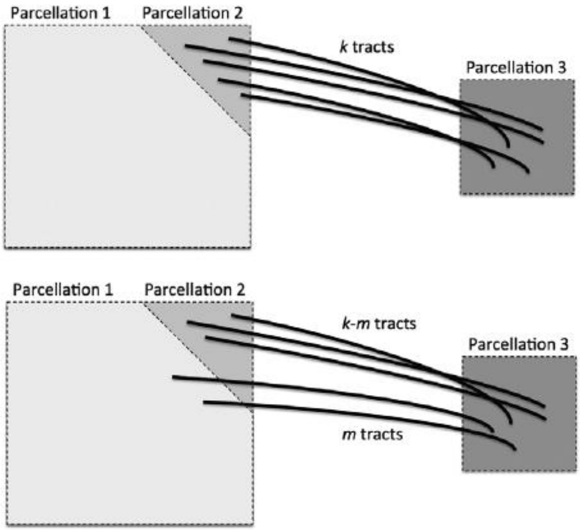


Figure 2.9 Schematic showing a possible tract mislabeling problem. k number of tracts are expected to pass through between parcellations 2 and 3 (top). However, m number of tracts might be mislabled to pass through parcellations 1 and 3 (bottom).

error in the tract count-based connectivity is then $(C_E - C_O)/C_E = m/k$. Let us determine the relative error for the tract length-based connectivity metric.

Suppose the ith tract has length L_i. The average tract length between the parcellations will be denoted as

$$\bar{L} = \frac{1}{k}\sum_{i=1}^{k} L_i.$$

The resistance of the ith tract is then given by

$$R_i = \bar{L} + \Delta L_i.$$

where $\Delta L_i = L_i - \bar{L}$ measures the difference from the mean. Subsequently, the expected total resistance is given by

$$\frac{1}{R_E} = \sum_{i=1}^{k} \frac{1}{\bar{L} + \Delta L_i} = \frac{1}{\bar{L}} \sum_{i=1}^{k} \Big(1 - \frac{\Delta L_i}{L} + \Big(\frac{\Delta L_i}{\bar{L}}\Big)^2 + \cdots\Big). \tag{2.6}$$

Since $\sum_{i=1}^{k} \Delta L_i = 0$, the expected total resistance is approximately

$$\frac{1}{R_E} \sim \frac{k}{\bar{L}} + \frac{1}{\bar{L}} \sum_{i=1}^{k} \Big(\frac{\Delta L_i}{\bar{L}}\Big)^2 \tag{2.7}$$

ignoring the cubic and other higher-order terms. Similarly, the observed resistance for $k - m$ tracts is given by

$$\frac{1}{R_O} = \sum_{i=1}^{k-m} \frac{1}{\bar{L} + \Delta L_i} \sim \frac{k-m}{\bar{L}} + \frac{1}{\bar{L}} \sum_{i=1}^{k-m} \Big(\frac{\Delta L_i}{\bar{L}}\Big)^2. \tag{2.8}$$

Hence the relative error of the new connectivity metric given by

$$\frac{1/R_E - 1/R_O}{1/R_E} \sim \frac{m + \sum_{i=k-m+1}^{k} \big(\Delta L_i/\bar{L}\big)^2}{k + \sum_{i=1}^{k} \big(\Delta L_i/\bar{L}\big)^2}.$$

The terms $\sum_{i=1}^{k} \big(\Delta L_i/\bar{L}\big)^2$ and $\sum_{i=k-m+1}^{k} \big(\Delta L_i/\bar{L}\big)^2$ are sufficiently small and the relative error is approximately m/k, which is the relative error in the tract count-based connectivity. Thus, we expect the error variability in the tract length-based connectivity to scale proportionally to that of the tract count-based method. So most likely all the methods will perform similarly in testing the connectivity differences at the edge level using the two-sample t-test since the t-test is scale invariant.

2.4 Probabilistic Connectivity

2.4.1 Diffusion

There are various probabilistic and stochastic models for tracing fibers (Basser and Pierpaoli, 1996; Hagmann et al., 2000; Batchelor et al., 2001; Behrens et al., 2007). Tench et al. (2002) introduced a hybrid streamline-based tractography where the direction of the principal eigenvector is modeled stochastically to overcome the shortcomings of DTI. Koch et al. (2002) introduced a Monte Carlo random walk simulation that uses a different transition probability than our own. Their algorithm has a certain restrictions built in the random walk so that it was only allowed to jump in a direction within 90 degrees from the previous jump direction, which restricts the jump to a very small number of voxels in the neighborhood. Furthermore, they considered the voxels with the FA-values (Basser and Pierpaoli, 1996) and sum of the eigenvectors bigger than certain thresholds. Then based on the Monte Carlo simulation of 4,000 random walks, they computed the probabilistic connectivity measure.

Hagmann et al. (2000) used a hybrid approach combining the Monte Carlo random walk simulation with information about the white fiber track curvature function in the corpus callosum. Then assuming bivariate normal distribution of the random walk hitting a vertical plane at some distance apart, they estimated the covariance matrix and performed a statistical hypothesis testing of the homogeneity of covariance matrix in the different regions of the corpus callosum.

Batchelor et al. (2001) solved an anisotropic heat equation where the diffusion coefficients of the heat equation are the diffusion coefficients of DTI. To get the probabilistic measure of the connectivity, the diffusion equation is solved with the initial condition where every voxel is zero except some seed region S where it is given the value 1:

$$\frac{\partial f}{\partial t}(p,t) = \nabla \cdot D\nabla f(p,t) \tag{2.9}$$

$$f(p,t=0) = 1 \text{ at } p \in S \text{ and } 0 \text{ elsewhere.} \tag{2.10}$$

The value 1 is then diffused though the white matter, and the numerical values between 0 and 1 are taken as a probability of white matter connectivity. Mathematically, it is equivalent as the Monte Carlo random walk simulation without restriction. The boundary condition can be also enforced. The boundary condition $D\nabla f \cdot \mathbf{n} = 0$ forces the boundary to be insulated, and no heat diffuses out of the boundary. The Crank–Nicholson scheme with Galerkin finite element discretization in space and finite difference in time was used to solve (2.10) (Babuška et al., 2004). Instead of solving the diffusion equation (2.10) directly, we can perform an equivalent iterative anisotropic kernel smoothing scheme (Chung et al., 2003b; Yoruk et al., 2005). Recently, fast marching-based tratography has been popularized (Parker et al., 2002; Staempfli et al., 2006; Jbabdi et al., 2008). Unlike probabilistic tractography, the fast marching methods do not present a computational burden.

The white fiber tracking is prone to cumulative acquisition noise and partial volume effect so the estimated white fiber tracks might possibly be erroneous in some cases (Basser et al., 2000; Tench et al., 2002). So it is crucial to develop a connectivity metric that is robust under the effect of acquisition noise and partial voluming. Such a robust seed-based probabilistic connectivity can be used in voxel-based morphometry (VBM) type of voxelwise inference on connectivity strength (Ashburner and Friston, 2000). In the classical VBM, the gray and white matter densities are computed and used for inference on tissue concentration at each voxel. In DTI, instead of the tissue densities, we can use the probabilistic connectivity that measures the strength of how two regions of the brain are connected.

2.4.2 Random Walk

In this section, a probabilistic connectivity based on the transition probability of diffusion is presented (Chung et al., 2003b; Yoruk et al., 2005). Let $P_t(p,q)$ be the *transition probability density* of a particle going from p to q under diffusion in time t. This is the conditional probability density of the particle hitting q at time t when the particle is at p at time 0. The *transition probability* of going from point p to another region of interest Q is given by

$$P_t(p,Q) = \int_Q P_t(p,x)\, dx,$$

where the integral is taken over every another voxel x.

Note that

$$P_t(p,\mathbb{R}^n) = \int_{\mathbb{R}^n} P_t(p,x)\, dx = 1.$$

The region Q can be a collection of voxels, and it may possibly be consisting of a single voxel p. So we will interchangeably use $P_t(p,q)$ as either transition probability density or transition probability if there is no ambiguity. The transition probability is the most natural probabilistic measure associated with diffusion. The connectivity measures based on the transition probability are naturally intuitive.

If the diffusion coefficient D is constant in $\mathbb{R}^n$, it can be shown that

$$P_t(p,q) = K_t(q-p) = \frac{1}{(4\pi t)^{n/2}(\det D)^{1/2}} \exp\Big(-\frac{x^\top D^{-1}x}{4t}\Big),$$

the Gaussian kernel (Stevens, 1995). Since D is varying over the brain regions, it is only valid when p and q are a short distance apart and we may take $D(x)$ to be constant in the neighborhood of voxel position x.

The transition probability of a particle going from p to any arbitrary q is the total sum of the probabilities of going from p to q through all possible intermediate points $x \in \mathbb{R}^n$. Therefore,

$$P_t(p,q) = \int_{\mathbb{R}^n} P_s(p,x)P_{t-s}(x,q)\, dx \tag{2.11}$$

for any $0 < s < t$. It is traditionally called the Chapman–Kolmogorov equation (Paul and Baschnagel, 1999). The equation still hold in the case when s is either 0 or t, since in that case one of the probability in the integral becomes the Dirac-delta function and, in turn, the integral collapses to the probability on the left side.

Note that the probability $P(p,x)$ decreases exponentially as the distance between p and x increases, so we approximate (2.11) in a small region B_p centered around p. For any point $x \in B_p$,

$$P_s(p,x) = K_s(p-x).$$

Then for any arbitrary points p and q,

$$P_t(p,q) = \frac{\int_{B_p} K_s(p-x)P_{t-s}(x,q)\,dx}{\int_{B_p} K_s(p-x)\,dx}. \tag{2.12}$$

When $s \to 0$, the approximation becomes exact since all the weights of the kernel will be in B_p. The denominator is a correction term for compensating the underestimation in the numerator. Note that this is the integral version of Gaussian kernel smoothing of data $P_{t-s}(x,q)$ for given q. Comparing with the formulation of Gaussian kernel smoothing, we rewrite (2.12) as

$$P_t(p,q) = \tilde{K}_s * P_{t-s}(p,q), \tag{2.13}$$

where the convolution is with respect to the first argument p and $\tilde{K}_s$ is the truncated Gaussian kernel normalized by $\int_{B_p} K_s(p-x)\,dx$. Note that when $s \to 0$, the equation becomes exact.

The kernel smoothing formulation (2.13) is mainly valid when s is small. For large s, we borrow the iterative kernel smoothing framework (Chung et al., 2005b). We discretize t into N equal time intervals $t = N\Delta t$ and let $s = \Delta t$. Then (2.13) can be written as

$$F_j(q) = \tilde{K}_{\Delta t} * F_{j-1}(q), \tag{2.14}$$

where $F_j(q) = P_{j\Delta t}(p,q)$ for a given p and the initial condition

$$F_0(q) = P_0(p,q) = \delta(p-q).$$

The reason we get the Dirac-delta function is that the transition probability of a particle at p hitting any point q instantaneously is zero except when $q = p$.

One important property of our iterative procedure is the conservation of the total probability at each iteration. From (2.14), we have

$$\begin{aligned}\int_{\mathbb{R}^n} F_{j+1}(x)\,dx &= \int_{B_x} \tilde{K}_{\Delta t}(x-y)\,dy \int_{\mathbb{R}^n} F_j(x)\,dx\\ &= \int_{\mathbb{R}^n} F_j(x)\,dx.\end{aligned}$$

Since

$$F_1(\mathbf{q}) = \tilde{K}_{\Delta t} * \delta(q) = \tilde{K}_{\Delta t}(q),$$

F_1 is a probability function and it will integrate to one so

$$\int_{\mathbb{R}^n} F_j(x)\, dx = 1.$$

Hence F_j is also a probability function at each iteration. As the number of iteration increases, the total probability will be dispersed over all regions of white matter from the seed.

If there are one million voxels within the brain, on average, each voxel will have the connection probability of one over a million, which is extremely small. So even though the connectivity measure based on the transition probability is a mathematically sound one, it may not be a good one for visualization. So what we need is the log scale of the transition probability, i.e. $\rho = \ln P_t(p,q)$, and we propose this as a probabilistic metric for measuring the strength of the anatomical connectivity. We will refer this metric as the *log-transition probability*. For simplicity, we may let $p = 0$ and let $\rho(q) = \ln P_t(0,q)$ for fixed t. If the diffusion coefficient is constant, the log-transition probability can be represented in a simple formula

$$\rho(x) = -x^\top D^{-1} x - \sum_{i=1}^{n} \ln \lambda_i - \frac{n}{2} \log(4\pi t),$$

where λ_i are the eigenvalues of D. When $D = I$,

$$\rho(x) = -x^\top x - \frac{n}{2} \log(4\pi t).$$

For a region of interest Q, the log-transition probability of reaching Q would be

$$\rho(q) = \ln \int_Q P_t(\mathbf{0}, x)\, dx.$$

2.5 Parcellation-Free Brain Network

In constructing connectivity matrices, brain parcellations are often used. However, there is no gold standard for parcellation, so the identification of node depends on the choice of parcellation. Depending on the scale of parcellation and the position of the parcellation, the topological properties of the graph varies considerably up to 95% (Fornito et al., 2010; Zalesky et al., 2010). Figure 2.10 illustrates how network changes over the use of different parcellations. A reasonable question to ask is whether it is possible to construct a connectivity matrix without the usual parcellation scheme. In this section, we present a parcellation-free, scalable, and iterative connectivity

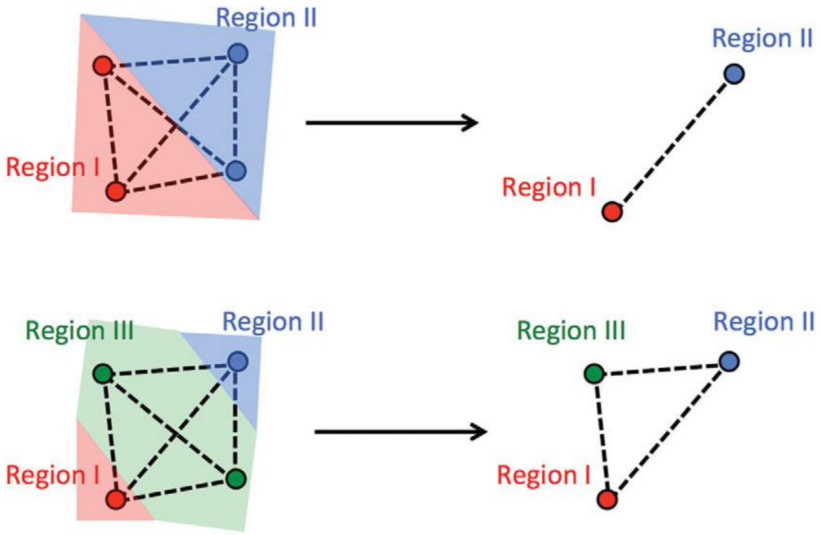


Figure 2.10 If different parcellations are used, we may end up with different networks. Four nodes are partitioned by three regions (top) resulting in a graph with with one edge and two nodes. Four nodes are partitioned by two regions (bottom) resulting in a graph with one cycle, three edges, and three nodes.

network construction technique called the ϵ*-neighbor construction* that avoids parcellation (Chung et al., 2011a,c; Lee et al., 2018b).

The ϵ-neighbor construction is motivated by the Rips complex of point cloud data (Ghrist, 2008), which is used to characterize the topology of the point cloud data. The Rips complex is a graph constructed by connecting two data points if they are within specific distance. The problem of the Rips complex is that given n data points, it exactly produce a graph with p nodes so the resulting graph becomes very dense when p becomes large. Unlike the Rips complex, the ϵ-neighbor method does not use every data point in constructing a graph, so it significantly reduces the complexity of the resulting graph. Further, while the point cloud data does not have any hidden topological constraint, the two end points of white matter fibers are connected so we are actually dealing with paired point cloud data. So the ϵ-neighbor construction is different from building the Rips complex while offering substantial computational advantage.

2.5.1 Epsilon Neighbor Construction

A graph G consists of a vertex set V and an edge set E, i.e., $G = \{V, E\}$. A point p is the ϵ-neighbor of G if the shortest distance between p and some point q in V is smaller than given ϵ. Then we identify the point p and q as the same point.

Definition 2.1 *The distance $d(p,G)$ of a point p to the graph G is the shortest distance between p and points in V, i.e.,*

$$d(p,G) = \min_{q \in V} \|p - q\|.$$

We say point p is the ϵ-neighbor *of graph G if $d(p,G) \leq \epsilon$.*

With these definitions, we construct a connectivity graph iteratively. In constructing the brain network, only two end points of the tract were considered since all other points along the tract are connected to these two points. We now construct the graph in an iterative fashion by adding one tract at a time to an existing graph. Initially graph G_1 consists of a single tract consisting of two end points e_{11} and e_{12}, and an edge $e_{11}e_{12}$ connecting the end points. Consider a tract with two end points. The algorithm then starts with the graph $G_1 = \{V_1, E_1\}$, where

$$V_1 = \{e_{11}, e_{12}\}, E_1 = \{e_{11}e_{12}\}.$$

In the next iteration, we consider how to add the second tract to the existing graph G_1 and obtain a new graph G_2. Consider the second tract with two end points e_{21}, e_{22} to the existing graph G_1. There are six possibilities in adding the two end points to G_1 depending if the end points are the ϵ-neighbors of G_1 (Figure 2.11) (Lee et al., 2018b):

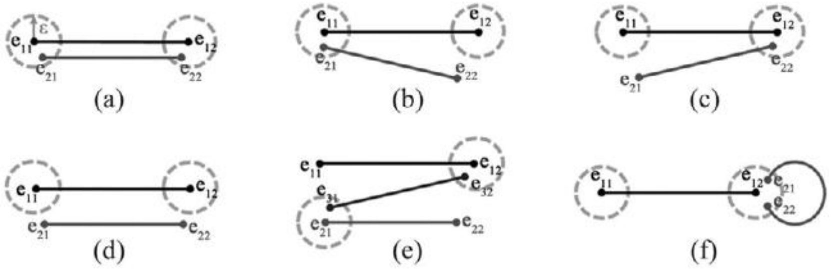


Figure 2.11 Six possibilities of the ϵ-neighbor construction: (a) e_{21} and e_{22} are all ϵ-neighbors of G_1. (b) Only e_{21} is an ϵ-neighbor of G_1. (c) Only e_{22} is an ϵ-neighbor of G_1. (d) Neither e_{21} nor e_{22} is an ϵ-neighbor of G_1. (e) e_{31} is an ϵ-neighbor of e_{21} in G_2 and e_{32} is an ϵ-neighbor of e_{12} in G_2. (f) e_{21} and e_{22} are ϵ-neighbors of e_{11} or e_{12} in G_1. This case is considered to be noise, because it resulted in a circular tract. The figure was generated by Min-Hee Lee of Yonsei University (Lee et al., 2018b).

1. e_{21} and e_{22} are all ϵ-neighbors of G_1. Since the end points e_{21} and e_{22} are close to the already existing graph G_1, we do not change the vertex set, i.e., $V_2 = V_1$. Now check if the edge $e_{21}e_{22}$ is in the edge set E_1 and add them if it is not found in the edge set. In this case, we have

$$E_2 = E_1 \cup \{e_{21}e_{22}\}.$$

2. Only e_{21} is an ϵ-neighbor. We only to add e_{22} to V_1 and let

$$V_2 = V_1 \cup \{e_{21}\},\ E_2 = E_1 \cup \{e_{21}e_{22}\}.$$

3. Only e_{22} is an ϵ-neighbor. We add e_{21} to V_1 and let

$$V_2 = V_1 \cup \{e_{22}\},\ E_2 = E_1 \cup \{e_{21}e_{22}\}.$$

4. e_{21} and e_{22} are not ϵ-neighbors. We add the end points to the vertex set and add the edge to the edge set. In this case, $e_{21}e_{22}$ forms a disjoint edge and we have

$$V_2 = V_1 \cup \{e_{21}, e_{22}\},\ E_2 = E_1 \cup \{e_{21}e_{22}\}.$$

Two other additional cases are explained in Figure 2.11 (Lee et al., 2018b). The procedure is iteratively performed to every tract until we exhaust all the tracts. The MATLAB code for the ϵ-neighbor construction is provided.[5]

The constructed 3D network's graph can be uniquely parameterized by transforming the graph into adjacent matrices (Figure 2.12). The adjacency matrix $A = (a_{ij})$ of a graph is constructed on the fly at each iteration by checking if we are adding a new edge to the existing edge set. If nodes i and j are connected, we let $a_{ij} = 1$ and $a_{ij} = 0$ otherwise. The diagonal terms a_{ii} are assumed to be zero. The adjacency matrix is symmetric. The adjacency matrix of a graph can be constructed on the fly at each iteration by checking if we are adding a new edge to the existing edge set. The adjacency matrix contains sufficient information to construct a graph. Statistical analysis can be done on the ensemble of adjacency matrices and we can determine if two groups significantly differ in connectivity.

Using the ϵ-neighbor construction used in building parcellation-free brain networks (Chung et al., 2011c), it is possible to build a filtration similar to the Rips filtration (Zomorodian and Carlsson, 2005; Edelsbrunner and Harer, 2008; Ghrist, 2008; Horak et al., 2009). Similar to the Rips filtration, the ϵ-neighbor filtration is a sequence of networks obtained from the ϵ-neighbor method. At the kth iteration, we have a network $\mathcal{G}_k$. As the number of iteration increases, we are generating a sequence of larger networks

$$\mathcal{G}_1 \subset \mathcal{G}_2 \subset \mathcal{G}_3 \subset \cdots,$$

[5] http://brainimaging.waisman.wisc.edu/~chung/graph/

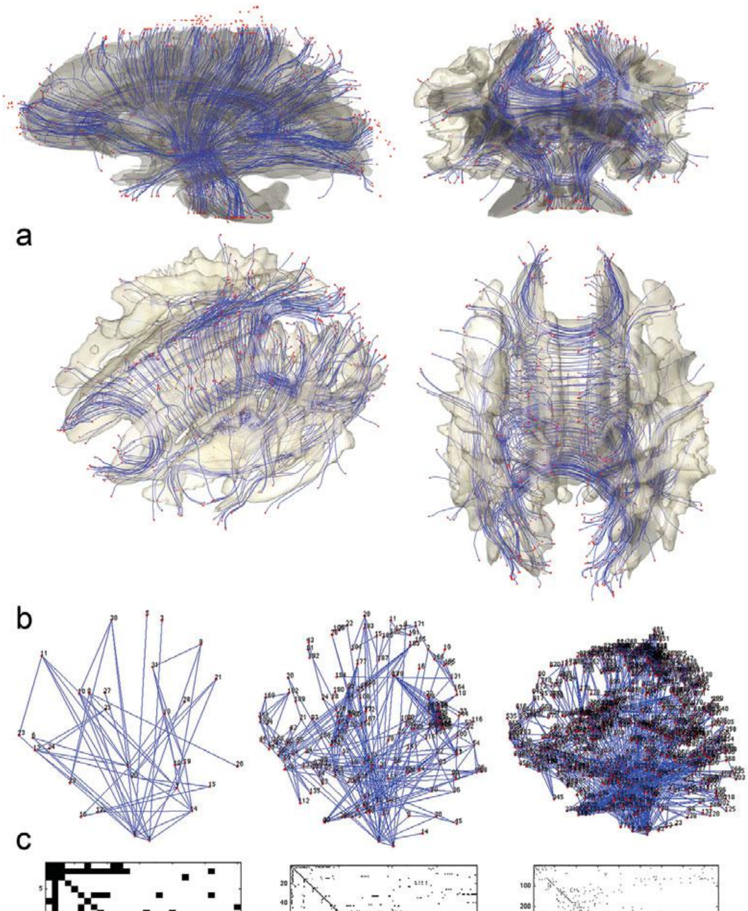


Figure 2.12 (a) White matter fibers are obtained from a streamline algorithm (Chung et al., 2017b). The end points are colored red. The surface is the isosurface of the FA-map, so some tracts are expected to be outside of the surface. The ϵ-neighbor method will use the proximity of the ends points in constructing the network graph. (b) The ϵ-neighbor graph construction on white matter fibers with 20, 10, and 6 mm radius. (c) Corresponding adjacency matrices.

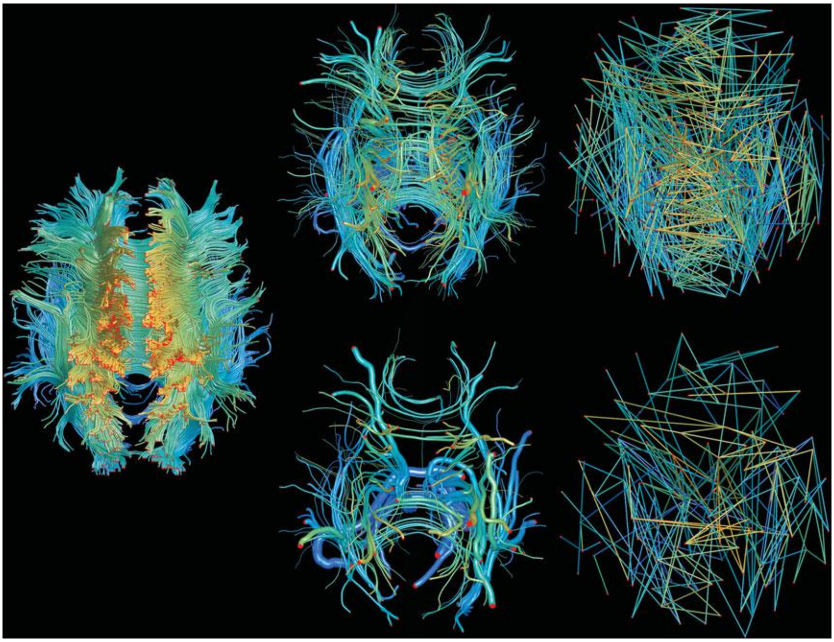


Figure 2.13 (a) The end points of tracts are identified. Tracts whose end points are within the ball of ϵ radius are connected. The fiber tracts and the balls can be viewed as a complex electronic circuit. (b) All the parallel circuits present in the system are shown. Each parallel circuit is replaced by a single tract with equivalent resistance. (c) The simplified circuit then forms a graph where the edge weights are given by the resistance.

which we call the ϵ-neighbor filtration, which is much easier to interpret since it shows the actual process of network construction. It is also possible to combine the electrical circuit model with the epsilon neighbor construction (Figure 2.13).

2.6 Structural Covariates

Traditionally structural brain networks are constructed using DTI. However, it is also possible to construct structural brain connectivity using T1-weighted MRI without DTI. Such an approach is usually called the *structural covariate* approach. It is originally based on the idea of correlating local morphological features obtained from MRI in constructing a structural brain network (Worsley et al., 2005a; Lerch et al., 2006; Chung et al., 2010c; Kim et al., 2011, 2012a). The previous works mainly focused on the cortico–cortical connectivity using cortical thickness, which is defined along the gray matter. Using the cross-correlation of cortical thickness, we then determine when the anatomy of one region changes, if there are corresponding morphological changes in other regions (Worsley et al., 2005a,b; Lerch et al., 2006). Cortical thickness was mainly chosen because it reflects the size, density, and arrangement of neurons (He et al., 2007). However, cortical thickness cannot be used in directly characterizing the connectivity within the white matter. To overcome the limitation of the previous studies, it is possible to correlate the Jacobian determinant obtained from the tensor-based morphometry (TBM) over different white matter voxels in determining association within the white matter as well (Chung et al., 2010c; Kim et al., 2011, 2012a).

2.6.1 Jacobian Determinants

TBM has been often used in characterizing tissue volume difference between populations at the voxel level (Chung et al., 2001b; Thompson et al., 2001). So far, most TBM studies have performed massive univariate tests in every voxel mainly using the Jacobian determinant. Such massive univariate approaches are ill suited for addressing more complex hypotheses about brain network connectivity. Most of structural brain network models involve DTIs in establishing edges in connectivity graphs (Hagmann et al., 2007; Bullmore and Sporns, 2009; Li et al., 2009; Zalesky and Fornito, 2009). Instead of using cortical thickness for constructing structural connectivity maps (Worsley et al., 2005a,b; Lerch et al., 2006; He et al., 2007), we can use voxelwise morphometric measures such as tissue density or the Jacobian determinant in building whole brain 3D connectivity maps. The main innovation is then the proposed framework, which does not utilize DTI but is still able to construct the population specific connectivity maps only using T1-weighted MRI.

TBM usually produces the displacement $u = (u_1, u_2, u_3)^\top$ of warping a template image to an individual subject image. With respect to the spatial coordinates $x = (x_1, x_2, x_3)^\top$, the displacement gradient tensor of u is given by (Ashburner et al., 2000; Chung et al., 2001b)

$$\nabla u = \frac{\partial u}{\partial x^\top} = \begin{pmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\ \frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3} \end{pmatrix}.$$

The nine components from scalar fields u measure the second-order morphological variabilities. The Jacobian matrix $J = (J_{ij}(x))$ is given by

$$J(x) = I + \nabla u,$$

where I denotes an identity matrix. In brain imaging, a voxel can be considered as the unit cube; therefore, $\frac{\partial J}{\partial t}(x)$ essentially measures the change in the volume of voxel x after the deformation. Expanding the Jacobian J, we get

$$\begin{aligned} J &= \det(I + \nabla u) \\ &= 1 + \mathrm{tr}(\nabla u) + \mathrm{detr}_2(\nabla u) + \det(\nabla u), \end{aligned}$$

where $\mathrm{detr}_2(\nabla u)$ is the sum of 2×2 principal minors of ∇U. For relatively small displacements, which is the case in brain development, we may neglect the higher-order terms and the Jacobian determinant is linearly approximated using the volume dilatation (Chung et al., 2001b):

$$\det J \approx 1 + \mathrm{tr}(\nabla u) = 1 + \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3}. \tag{2.15}$$

In elastic theory, the volume dilatation is defined as (Marsden and Hughes, 1983)

$$\nabla \cdot u = \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3}.$$

When $\nabla \cdot u(x) = 0$ in the neighborhood of x, the deformation is incompressible so there is no volume change. However, if $nabla \cdot U(x) > 0$, and the volume increases while $\nabla \cdot u(x) < 0$, the volume decreases after the deformation.

2.6.2 Correlating Jacobian Determinants

Seed-based connectivity. We can correlate the Jacobian determinants over the whole brain by fixing one voxel. Figure 2.14 displays the pipeline for the popular seed-based connectivity analysis. We illustrate the method with MRI of postinstitutionalized (PI) children, where a seed voxel is identified at the genu of the corpus callosum. The connectivity maps were computed for normal controls and PI. The regions of high connectivity are shown in darker red while the regions of low connectivity are shown in lighter red. Since the seed is taken at the genu, white matter regions near the genu should have higher connectivity. The connectivity maps look very similar to probabilistic connectivity maps often obtained in DTI. One can actually trace the gradient of connectivity maps by solving the streamline equation and obtain tracts that should behave like white matter fiber tracts.

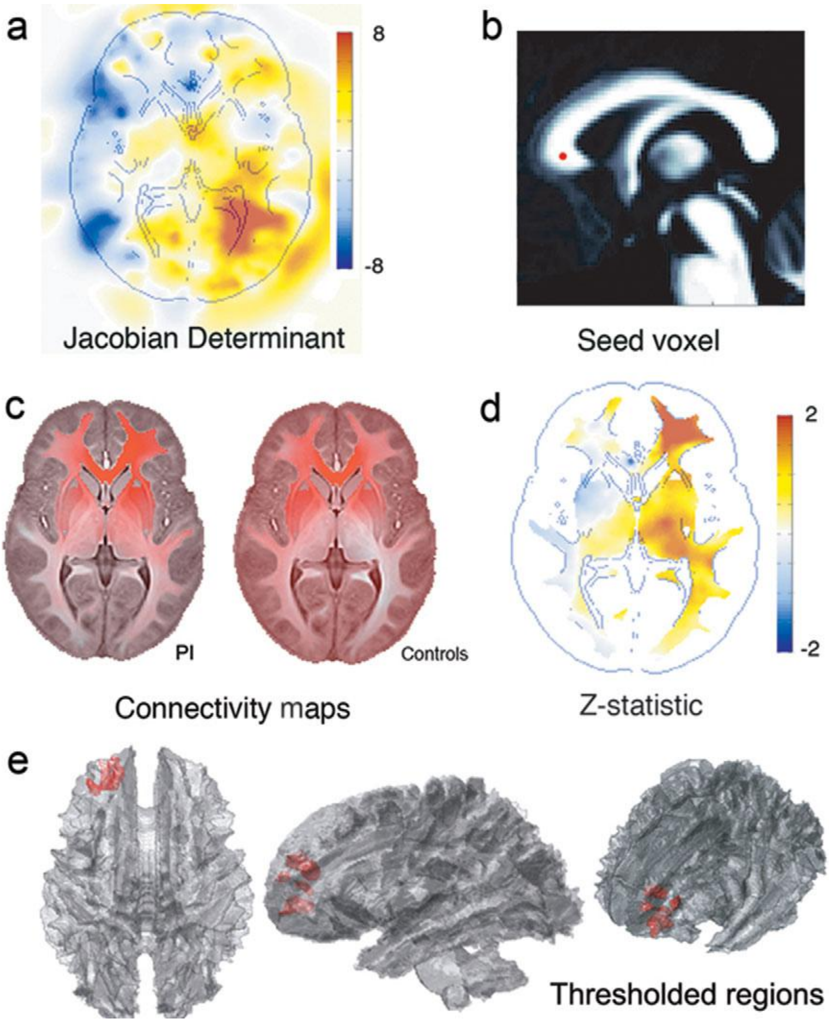


Figure 2.14 Seed-based connectivity. (a) Jacobian determinant map from a subject to a template. (b) Seed voxel is chosen at the genu of the corpus callosum. (c) Connectivity maps computed from the seed for postinstitutionalized (PI) children and controls. (d) Z-statistic showing the connectivity difference. (e) Region of significant connectivity difference thresholded at $p = 0.01$.

From the connectivity maps, Z-statistic is constructed by the Fisher transform. The regions of the most significant connectivity difference are localized by thresholding the p-values of the Z-statistic (PI - controls) at $p = 0.01$ and overlaid on the white matter boundary. The affected regions are white matter regions connecting the anterior prefrontal cortex. We mainly observe the highly clustered regions of positive correlation difference only. Increase in the white matter volume in the genu corresponds to increase of white matter in the anterior prefrontal cortex, indicating the abnormal corpus callosum connectivity in PI. However, this does not imply that PI has more white matter volume in these regions.

Cross-correlation. Instead of seed-based connectivity, which fixes one node, we can compute the cross-correlation ρ of the Jacobian determinants between different voxels as a way to establish structural connectivity. Since correlation is invariant under translation, the constant 1 in (2.15) does not contribute to the cross-correlation computation. Therefore, if $K = (K_{ij})$ is another Jacobian matrix at a different voxel position, the cross-correlation ρ between the Jacobian determinants is approximated as

$$\rho(trJ, trK) = \sum_{i,j} \rho(J_{ii}, K_{jj}).$$

By fixing one region to be a seed, we can construct a correlation map that looks similar to seed-based probabilistic connectivity maps in DTI (Batchelor et al., 2001; Koch et al., 2002). The resulting correlation map measures the strength of connection from the seed to other white matter regions. Statistical analysis on connectivity map difference can be done using the Fisher transform. Given two connectivity maps ρ_1 and ρ_2, we can construct the Z-statistic using the Fisher transform

$$Z = c(\tanh^{-1} \rho_1 - \tanh^{-1} \rho_2)$$

with some normalizing constant c.

2.6.3 Correlating Tensors via RV-Coefficients

The problem of using the Jacobian determinant in constructing the connectivity map is that it summarizes a 3 × 3 matrix into a single scalar value so we are not utilizing the full multivariate information. We need a more general approach in correlating matrices using a multivariate extension of correlation. Instead of correlating Jacobian determinants, we can correlate Riemmanian metric tensors induced by the Jacobian matrix (Chung et al., 2008b). The Jacobian matrix J induces the Riemannian metric tensor

$$g = (g_{ij}) = J^\top J.$$

Then we can use the induced metric tensor instead of the Jacobian determinant in the log-Euclidean framework (Arsigny et al., 2005; Lepore et al., 2006).

Note that the volume element $\sqrt{\det g}$ is identical to the Jacobian determinant $\det J$.

Since the metric tensor g is symmetric and positive definite, we can use the *RV-coefficient* (Robert and Escoufier, 1976; Shinkareva et al., 2006; Abdi, 2007).

Definition 2.2 *The RV-coefficient ρ between two positive definite symmetric matrices g and h is defined as*

$$\rho(g,h) = \frac{tr(gh)}{\sqrt{tr(g^2)\, tr(h^2)}}.$$

The trace operator tr is equivalent to the vector product if we vectorize the entries of matrices and $tr(g^2)$ is the square of the Frobenius norm of g. If $h = K^\top K$, the numerator represents a generalized covariance between J and K while the denominator normalized the numerator to have values between 0 and 1. Unfortunately, the statistical distribution of the RV-coefficient is unknown analytically but the exact permutation distribution can be obtained so that the exact mean and variance can be computed (Kazi-Aoual et al., 1995; Abdi, 2007). We define a sphericity index of tensor g to be

$$\beta_g = \frac{(trg)^2}{tr(g^2)}.$$

Note that the sphericity index was mainly used in estimating the degrees of freedom for multivariate tests (Worsley et al., 1995). Then the mean of RV-coefficient between tensors g and h is

$$\mathbb{E}(\rho) = \frac{\beta_g \beta_h}{n-1},$$

where n is the dimension of matrices. The mean is taken over all possible permutations between g and h. The computation for the variance of RV-coefficient is more complicated.

$$\mathbb{V}(\rho) = \frac{2(n-1-\beta_g)(n-1-\beta_h)}{(n+1)(n-1)^2(n-2)}\left[1 + \frac{n-3}{2n(n-1)}\gamma\right]$$

for some γ. For $n = 3$, we do not need to know the value of γ since the term vanishes in the expression.

The Z statistic for testing the significance of correlation map difference between two RV-coefficients ρ_1 and ρ_2 is then given by

$$Z = \frac{\rho_1 - \mathbb{E}\rho_1 - (\rho_2 - \mathbb{E}\rho_2)}{\sqrt{\mathbb{V}\rho_1 + \mathbb{V}\rho_2}}.$$

which follows the standard normal distribution.

3

Graph Theory

Recent developments in graph theoretic analysis of complex networks have led to deeper understanding of brain networks. Many complex networks show similar macroscopic behaviors despite differences in the microscopic details (Bullmore and Sporns, 2009). Probably the two most often observed characteristics of complex networks are scale-free and small-world properties (Song et al., 2005). In this chapter, we will explore whether brain networks follow scale-free and small-worldness among other graph theory properties.

3.1 Trees and Graphs

Many objects and data can be represented as networks. Unfortunately, networks can be very complex when the number of nodes increases. Trees, which can be viewed as the backbone of the networks, are often used as a simpler representation of the networks.

Definition 3.1 *An (undirected)* graph *is an ordered pair $G = (V, E)$ with a node (vertex) set V and edge (link) set E connecting the nodes. The vertices belonging to an edge are called the ends or end vertices of the edge. A vertex may exist in a graph and not belong to an edge. The size of the graph is usually determined by the number of vertices $|V|$ and the number of edges $|E|$.*

A (weighted) graph with p nodes is often represented as a connectivity matrix $C = (c_{ij})$ of size $p \times p$, where c_{ij} are usually referred to as edge weights. A *binary graph* is a graph with binary edge weights, i.e., $(0, 1)$. An undirected graph yields a symmetric connectivity matrix. For instance, normalized and scaled data matrix X, Y of size $n \times p$, $corr(X, X)$ gives an undirected weighted graph while $corr(X, Y)$ gives a directed weighted graph.

Definition 3.2 *A* tree *is an undirected graph, in which any two nodes are connected by exactly one path. Nodes with only one edge in a tree are referred to as* leaves *or* leaf nodes *(Stam et al., 2014). The number of leaves is the* leaf number. *A* binary tree *is a tree, in which each node has at most two connecting nodes called left or right children. A* forest *is a disjoint union of trees.*

In brain imaging, we often deal with brain cortical mesh vertices. To distinguish such vertices from the vertices of graphs, we will simply use the term *nodes*. Instead of edges, the term *links* is also used. The term *tree* was coined in 1857 by the British mathematician Arthur Cayley. Node degrees are possibly the most often used feature in characterizing the topology of a tree. In a tree, there is only one path connecting any two nodes. Thus, the average *path length* between all node pairs is easy to compute.

Theorem 3.1 *For a tree, we have* $|V|-|E|=1$. *For a forest, the total number of trees is given by* $|V|-|E|$.

The proof is straightforward enumeration of edges with respect to the nodes.

Definition 3.3 *A* rooted tree *is a tree, in which one node is designated the root. In a rooted tree, the* parent *of a node is the node connected to it on the path to the root. Given node* x*, a* child *of* x *is a node of which* x *is the parent. A* sibling *to node* x *is any other node on the tree that has the same parent as* x.

Definition 3.4 *In a rooted tree, the* ancestors *of a node are the nodes in the path from the node to the root, excluding the node itself and including the root. The* descendants *of node* x *are those nodes that have* x *as an ancestor.*

Definition 3.5 *The* height *of a node in a rooted tree is the length of the longest downward path from a leaf from that vertex. The* depth *of a node is the length of the path to its root.*

We can show that every node except the root has a unique parent. The root has depth zero, leaves have height zero.

3.2 Minimum Spanning Trees

The trees are mainly used as a simpler representation of more complex graphs. The computation of graph theory features for trees is substantially easier that that of graphs. Among all trees, minimum spanning trees (MST) are most often used in practice.

Definition 3.6 *A* spanning tree *of a graph G is a tree whose node set is identical to the node set of G. The* minimum spanning tree *(MST) is a spanning tree whose sum of edge weights is the smallest.*

Theorem 3.2 *If all the edge weights are unique in a graph, there exists one unique MST.*

Proof. We prove by contradiction. Suppose there are two MSTs $M_1 = (V, E_1)$ and $M_2 = (V, E_2)$. Since $E_1 \neq E_2$, there are edges that belongs to one but not both. Among such edges, let e_1 be the one with the least weight. Without loss of generality, assume $e_1 \in E_1$. Adding e_1 to M_2 will create a cycle C. Since M_1 has no cycle, C must have an edge e_2 not in M_1, i.e., $e_2 \in E_2$. Since e_1 is the least edge weight among all the edges that belong to one but not both, the edge weight of e_1 is strictly smaller than the weight of e_2. Replacing e_2 with e_1 will yield a new spanning tree with weights less than that of M_2, which is a contradiction. Thus, there must be only one unique MST. □

MST is often constructed using Kruskal's algorithm (Lee et al., 2012). Kruskal's algorithm is a greedy algorithm with run time $O(|E| \log |E|) = O(|E| \log |V|)$. Note $|E|$ is bounded by $|V|(|V| - 1)/2$. The run time is equivalent to the run time of sorting the edge weights, which is also $O(|E| \log |E|)$. The algorithm starts with an edge with the smallest weight. Then add an edge with the next smallest weight. This sequential process continues while avoiding a loop and generates a spanning tree with the smallest total edge weights (Figure 3.1). Thus, the edge weights in MST correspond to the order, in which the edges are added in the construction of MST. It is known that the single linkage hierarchical clustering is related to Kruskal's algorithm of MST (Gower and Ross, 1969).

Consider two graphs C_1 and C_2 shown in Figure 3.1. In Matlab, the edge weights can be encoded as weighted adjacency matrices

```
C1= [0 0.2 0.7 0
      0.2 0 0.5 0.7
      0.7 0.5 0 0.5
      0 0.7 0.5 0]

C2= [0 0.2 0 0
      0.2 0 0.5 0.4
      0.7 0.5 0 0.7
      0 0 0.4 0]
```

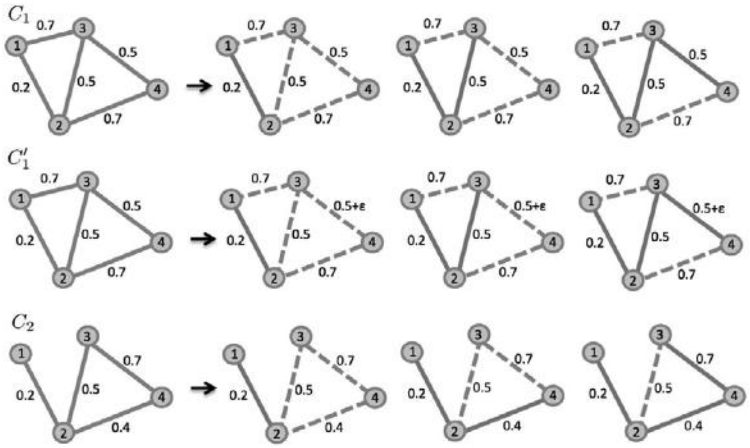


Figure 3.1 MST construction using Kruskal's algorithm. C_1: The edge weights of MST are 0.2, 0.5, and 0.5. Even though we have identical edge weights of 0.5, we can numerically still have a unique MST by putting infinitesimally small weight ϵ, which results in C_1'. C_2: The edge weights of MST are 0.2, 0.4, and 0.7.

The diagonals are assigned value zero. The minimum spanning tree is obtained using `graphminspantree.m`, which inputs a sparse connectivity matrix.

```
C1=sparse(C1)
C2=sparse(C2)
[treeC1,pred]=graphminspantree(C1)
[treeC2,pred]=graphminspantree(C2)

treeC1 =
   (2,1)        0.2000
   (3,2)        0.5000
   (4,3)        0.5000

 treeC2 =
   (2,1)        0.2000
   (3,2)        0.5000
   (4,3)        0.4000
```

By connecting all the nonzero entries in the output `treeC1` and `treeC2`, which are given as sparse matrices, we obtain MST.

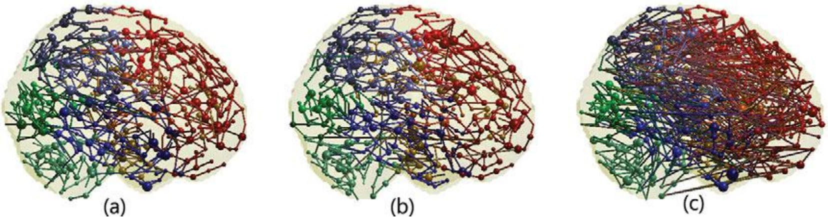


Figure 3.2 MST of the brain networks of (a) attention-deficit hyperactivity disorder, (b) autism spectrum disorder, and (c) pediatric control subjects. The number of nodes are 1,056 in all networks. The color represents the 97 regions of interest based on the predefined anatomical parcellations. Figure was generated by Hyekyoung Lee of Seoul National University (Lee et al., 2012).

When some edges have equal weight as shown in Figure 3.1, we may not have a unique MST. Thus, there are at most $q!$ possible MSTs. Given a binary graph with p nodes, any spanning tree is an MST. Given a weighted graph with q identical edge weights, we can assign infinitesimally small weights ϵ, 2ϵ, $\ldots, q\epsilon$ to the identical edge weights. This results in a graph with unique edge weights and there exists a single MST. There are at most $q!$ ways to assign infinitesimally weights to q edges. For a complete graph with p nodes, the total number of spanning tree is p^{p-2}. For an arbitrary binary graph, the total number can be calculated in polynomial time as the determinant of a matrix from Kirchhoff's theorem (Chaiken and Kleitman, 1978).

Theorem 3.3 *Suppose $\rho_1, \ldots, \rho_{p-1}$ are the ordered edge weights of the MST of a graph G. If $\rho'_1, \ldots, \rho'_{p-1}$ are the ordered edge weights of any other spanning tree of G. Then we have*

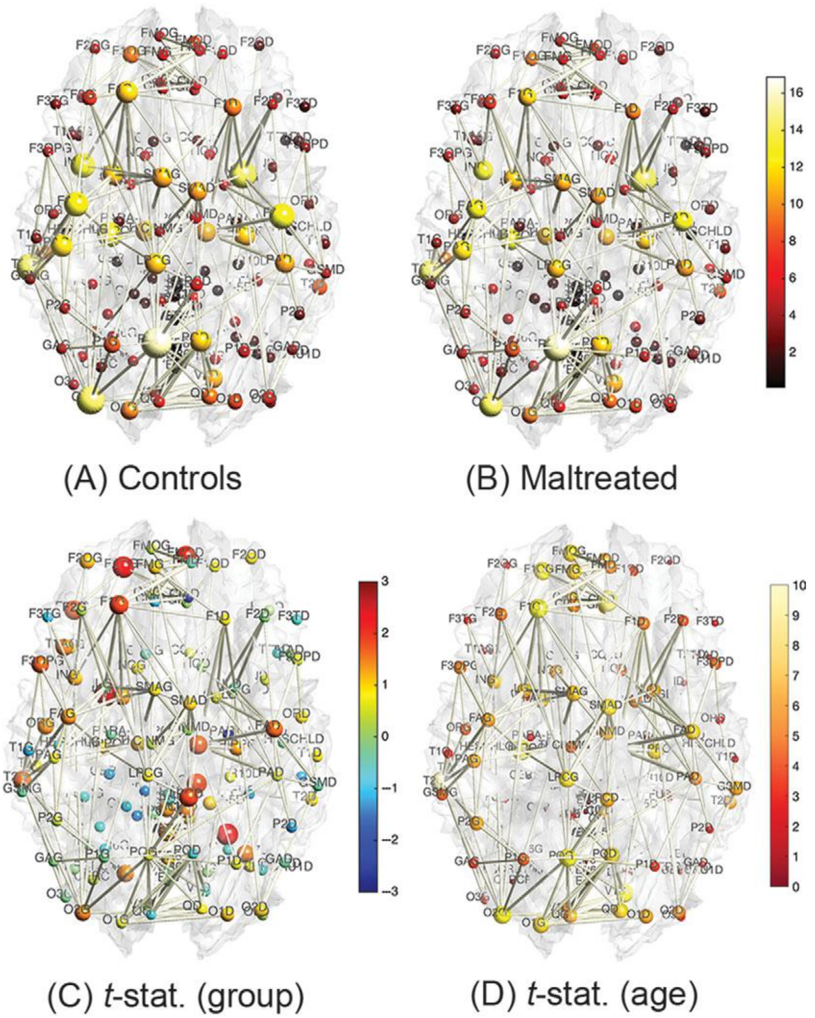
$$\rho_j \leq \rho'_j$$

for all j.

Theorem 3.3 shows the ordered edge weights of MST are extremely stable relative to any ordered edge weights of a spanning tree, and thus they can be used as a stable multivariate feature for quantifying graphs (see Figure 3.2).

3.3 Node Degree

Probably the most important network complexity measure is node degree. Many other graph theory measures are related to node degree (Bullmore and


(A) Controls

(B) Maltreated

(C) *t*-stat. (group)

(D) *t*-stat. (age)

Figure 3.3 The node size and color correspond to the mean degree of normal controls (a) and maltreated children (b) (Chung et al., 2017b). The edges are the average of the mean degrees of the two nodes thresholded at 0.5. (c) The two-sample *t*-statistic of the degree differences (controls – maltreated). (d) *t*-statistic of age effect while accounting for sex and group variables.

Sporns, 2009). The *degree k* of a node is the number of edges connected to it. It measures the local complexity of network at the node (Figure 3.3). Once we have the adjacency matrix of a network, the node degree can be computed easily by summing up the corresponding rows or columns in the adjacency matrix. The average node degree is simply given in terms of $|E|$ and $|V|$.

Theorem 3.4 *For a undirected network, the average degree* $\mathbb{E}k$ *is*

$$\mathbb{E}k = \frac{2|E|}{|V|},$$

where $|V|$ *is the total number of nodes and* $|E|$ *is the total number of edges in the graph.*

Proof. The total number of the node degree is $2|E|$. Thus, the average node degree is $2|E|/|V|$. □

3.3.1 Scale-Free Networks

The *degree distribution* $P(k)$, probability distribution of the number of edges k in each node, does not have heavy tails. The usual two-sample t-tests will not work in the tail regions since the variance is too large due to small sample size. Inference on tail regions requires extreme value theory, which deals with modeling extreme events and has seen applications in environmental studies (Smith, 1989) and insurance (Embrechts et al., 1999). One main tool in extreme value theory is the use of generalized Pareto distribution in approximating the tail distributions at high thresholds. A standard technique is to estimate tail regions with parametric models and perform inferences on the parameters of the model fit. For low degrees, since the sample size is usually large, a two-sample t-test is sufficient.

The degree distribution $P(k)$ can be represented by a power law with a degree exponent γ usually in the range $2 < \gamma < 3$ for diverse networks (Song et al., 2005; Bullmore and Sporns, 2009):

$$P_p(k) \sim k^{-\gamma}.$$

Such networks exhibit gradual decay of tail regions (heavy tail) and are said to be *scale-free*. In a scale-free network, a few hub nodes hold together many nodes, while in a random network, there are no highly connected hub nodes. The smaller the value of γ, the more important the contribution of the hubs in the network.

Previous studies have shown that the human brain network is not scale-free (Hagmann et al., 2008; Gong et al., 2009; Zalesky et al., 2010). Hagmann et al. (2008) reported that degree decayed exponentially, i.e.,

$$P_e(k) \sim e^{-\lambda k},$$

where λ is the rate of decay (Fornito et al., 2016). The smaller the value of λ, the more important the contribution of the hubs in the network.

Gong et al. (2009) and Zalesky et al. (2010) found the degree decayed in a heavy-tailed manner following an exponentially truncated power law

$$P_{etp}(k) \sim k^{-\gamma} e^{-\lambda k},$$

where $1/\lambda$ is the cutoff degree at which the power law transitions to an exponential decay (Fornito et al., 2016). This is a more complicated model than the previous two models.

The estimated best model fit can be further used to compare the model fits among the three models. However, existing literature on graph theory features mainly deal with the issue of determining if the brain network follows one of the aforementioned laws (Hagmann et al., 2008; Gong et al., 2009; Zalesky et al., 2010; Fornito et al., 2016). However, such model fit was not often used for actual group-level statistical analysis.

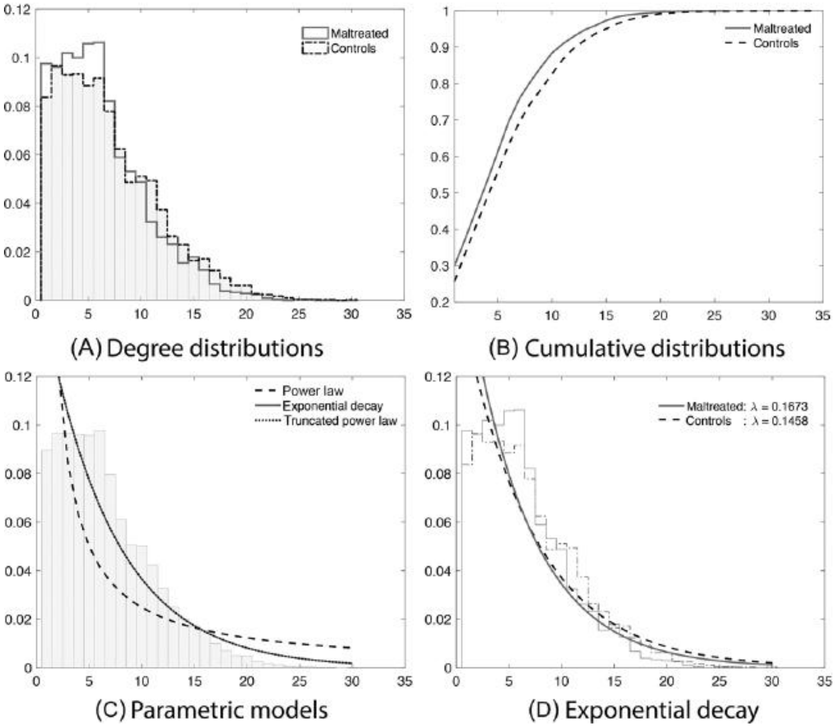
3.3.2 Estimating Degree Distributions

Directly estimating the parameters from the empirical distribution is challenging due to small sample size in the tail region. This is probably one of the reasons we have conflicting studies. To avoid the issue of sparse sampling in the tail region, the parameters are often estimated from cumulative distribution functions (CDF) that accumulate the probability from low to high degrees and reduce the effect of noise in the tail region. To increase the sample size further in the tail region, we combine all the degrees across the subjects. For the exponentially truncated power law, for instance, the two parameters γ, λ are then estimated by minimizing the sum of squared errors (SSE) using the L_2-norm between theoretical CDF F_{etp} and empirical CDF $\widehat{F}_{etp}$:

$$(\widehat{\gamma}, \widehat{\lambda}) = \arg \min_{\gamma \geq 0, \lambda \geq 0} \int_0^\infty \left| F_{etp}(k) - \widehat{F}_{etp}(k) \right|_2^2 dk.$$

We propose a two-step procedure for fitting node degree distribution. The underlying assumption of the two-step procedure is that each subject follows the same degree distribution law but with different parameters. In the first step, we need to determine which law the degree distribution follows at the group level. This is done by pooling every subject to increase the robustness of the fit. In the second step, we determine subject-specific parameters.

Step 1. This is a group-level model fit. Figure 3.4(a) shows the degree distributions of the combined subjects in a group (Chung et al., 2017b). To determine if the degree distribution follows one of the three laws, we combine


(A) Degree distributions

(B) Cumulative distributions

(C) Parametric models

(D) Exponential decay

Figure 3.4 (a) Degree distributions of all the subjects combined in each group (Chung et al., 2017b). (b) The cumulative distribution functions (CDF) of all the subjects in each group. (c) Three parametric model fit on the CDF of the combined 54 subjects. (d) The exponential decay model is fitted in each group. The estimated parameters are significantly different (p-value < 0.02).

all the degrees across subjects in the group. Since high-degree hub nodes are very rare, combining the node degrees across all subjects increases the robustness of the fit. This results in much more robust estimation of degree distribution. At the group-level model fit, this is possible.

Step 2. This is the subject-level model fit. Once we determine that the group follows a specific power law, we fit the same model for each subject separately.

3.3.3 Hub Nodes

Hubs or hub nodes are defined as nodes with a high degree of connections (Fornito et al., 2016). Table 3.1 shows the list of the 13 most connected nodes in a DTI study of maltreated children versus normal controls. The numbers

Table 3.1. *Thirteen most connected hub regions obtained from a DTI study comparing normal controls and maltreated children (Chung et al., 2017b). AAL regions are sorted in the descending order of the node degree. The controls have more connections without an exception compared to maltreated children.*

Label	Parcellation name	Combined	Controls	Maltreated
PQG	Precuneus-L	16.11	16.87	15.09
NLD	Putamen-R	14.96	15.26	14.57
O2G	Occipital-Mid-L	14.44	15.52	13.00
T2G	Temporal-Mid-L	14.30	15.16	13.13
HIPPOG	Hippocampus-L	13.15	13.94	12.09
FAD	Precentral-R	12.85	14.00	11.30
ING	Insula-L	12.56	13.61	11.13
FAG	Precentral-L	12.43	13.45	11.04
PQD	Precuneus-R	12.00	12.03	11.96
PAG	Postcentral-L	11.89	12.52	11.04
NLG	Putamen-L	11.39	11.68	11.00
F1G	Frontal-Sup-L	11.22	12.13	10.00
HIPPOD	Hippocampus-R	11.15	11.90	10.13

are the average node degrees in each group. All 13 nodes showed higher-degree values in the controls without an exception. The probability of this event happening by random chance alone is $2^{-13} = 0.00012$. This is an unlikely event and we conclude that the controls are more highly connected in the hub nodes compared to the maltreated children.

3.4 Shortest Path Length

Definition 3.7 *In a weighted graph, the* shortest path *between two nodes in a graph is a path whose sum of the edge weights is minimum. The* path length *between two nodes in a graph is the sum of edge weights in a shortest path connecting them (Bullmore and Sporns, 2009).*

When the weighted graph becomes binary, the path length is the number of edges in the shortest path. The shortest path is traditionally computed using Dijkstra's algorithm, which is a greedy algorithm with run time $O(|V|^2)$ invented by E. W. Dijkstra. The algorithm builds a *shortest path tree* from a root node, by building a set of nodes that have minimum distance from the root. The algorithm requires two sets, $\mathcal{S}$ and $\mathcal{N}$. $\mathcal{S}$ contains nodes included in the shortest path tree, and $\mathcal{N}$ contains nodes not yet included in the shortest path tree. At every step of the algorithm, we find a node that is in $\mathcal{N}$ and has minimum distance from the root.

Functional integration in the brain is the ability to combine information from multiple brain regions (Lee et al., 2018b). A measure of this integration is often based on the concept of path length. Path length measures the ability to integrate information flow and functional proximity between pairs of brain regions (Sporns and Zwi, 2004; Rubinov and Sporns, 2010). When the path length becomes shorter, the potential for functional integration increases.

3.5 Clustering Coefficient

Clustering coefficient describes the ability for functional segregation and efficiency of local information transfer.

The clustering coefficient of a node measures the propensity of pairs of nodes to be connected to each other if they are connected to another node in common (Watts and Strogatz, 1998; Newman et al., 2001). There are two different definitions of the clustering coefficients, but we will use the one originally given in Watts and Strogatz (1998).

Definition 3.8 *The clustering coefficient c_p at node p is a fraction of the number of existing connections between the neighbors of the node divided by the number of all possible connections of the graph.*

Let k_p be the node degree at p. At most, $k_p(k_p-1)/2$ edges can exist among k_p neighbors if they are all connected to each other. The *clustering coefficient* c_p of the node p is the fraction of allowable edges that actually exists over the theoretical limit $k_p(k_p - 1)/2$, i.e.,

$$c_p = \frac{\text{number of edges among neighbors of } p}{k_p(k_p - 1)/2}.$$

The *overall* clustering coefficient of a graph G, $c(G)$, is simply the average of the clustering coefficient c_p over all nodes, i.e.,

$$c(G) = \frac{1}{|V|} \sum_{p \in V} c_p,$$

where $|V|$ is the total number of nodes in the graph. Note that $0 \leq c(G) \leq 1$.

Random graphs are expected to have a smaller clustering coefficient compared to more structured one (Sporns and Zwi, 2004). For a complete

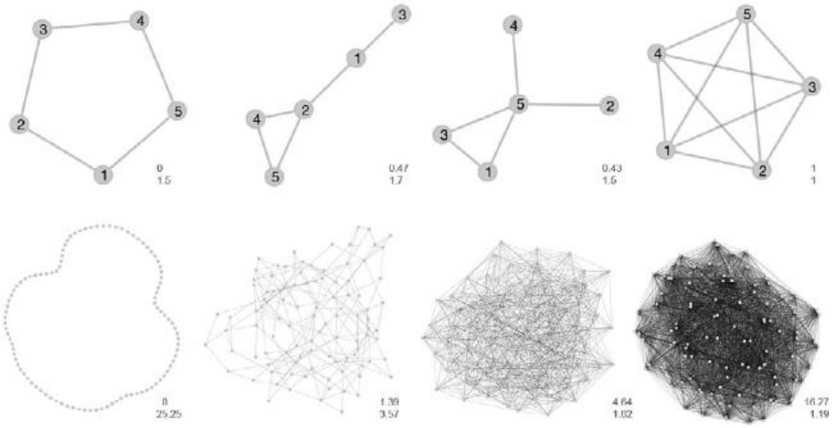


Figure 3.5 The numbers are the average clustering coefficients (top) and average path lengths (bottom). More complete the network becomes, it has shorter average path length and larger clustering coefficients.

graph, where all nodes are connected to each other, $c(G)$ obtains the maximum 1 and tends to zero for a random graph as the graph becomes large (Figure 3.5) (Newman et al., 2006). Definition 3.8 is biased for a graph with low degree nodes due to the factor $k_p(k_p - 1)$ in the denominator. Unbiased definition of the clustering coefficient is computed by counting the total number of paired nodes and dividing it by the total number of such pairs that are also connected. Compared to random networks, the brain network is known to have a higher clustering coefficient and shorter path length. These are the characterization of *small-world networks* (Sporns and Zwi, 2004).

3.6 Small-Worldness

If most nodes can be connected in a very small number of paths, the network is said to be *small-world*. Small-world networks have dense short-range connections with a relatively small number of long-range connections (Watts and Strogatz, 1998). The small-worldness of networks is usually defined with clustering coefficient $c(G)$ and average path length $\mathbb{E}l$.

Let l be the shortest path between two nodes, and let $|V|$ be the number of nodes in a graph. If we take the average of l for every pair of nodes and over all realizations of the randomness in the model, we have the mean path length $\mathbb{E}l$. The mean path length $\mathbb{E}l$ measures the overall navigability of a network.

$\mathbb{E}l$ is related to the diameter of the network, which is the maximum l (longest path) (Newman, 2003).

A network G is considered to be a small-world network if it meets the following criteria:

$$\gamma = \frac{c(G)}{c(R)} \gg 1$$
$$\lambda = \frac{\mathbb{E}l(G)}{\mathbb{E}l(R)} \approx 1$$
$$\sigma = \frac{\gamma}{\lambda} > 1,$$

where R denotes a random network that preserves the number of nodes, edges, and node degree distributions present in G. Often hundreds of random networks needed to be simulated for each network G to obtain stable estimates for the average path lengths and clustering coefficient. Network small-worldness is often quantified by ratio σ.

For regular lattices, $\mathbb{E}l$ scales linearly with the number of nodes $|V|$, while for random graphs, $\mathbb{E}l$ is proportional to $\ln|V|$ (Watts and Strogatz, 1998; Newman and Watts, 1999). The small-world network is somewhere between regular lattice and random graphs. Therefore, the small-worldness is mathematically expressed as follows (Song et al., 2005):

$$\mathbb{E}l \sim \ln|V|. \tag{3.1}$$

The relation (3.1) links over the size of the graph to the number of nodes and implies that as $|V|$ increases, the average path length is bounded by the logarithm of $|V|$. The model (3.1) can be rewritten as

$$|V| \sim e^{\mathbb{E}l}.$$

The relation (3.1) implies that the small-world networks are not self-similar, since self-similarity requires a power-law relation between l and $|V|$. However, Song et al. (2005) was able to show that the model (3.1) might be biased for inhomogenous networks. Using a scale-invariant renormalization procedure through the box counting method, Song et al. (2005) was able to show that diverse, complex networks are in fact self-similar.

3.7 Fractal Dimension

The fractal dimension is often used to measure self-similarity. Benoit B. Mandelbrot named the term *fractal* in 1960 (Mandelbrot, 1982). Mathematically, a fractal is a set with a nonintegral Hausdorff dimension (Hutchinson, 1981). While classical geometry deals with objects with an integer dimension, fractals have a nonintegral dimension. Fractals have infinite details at all points of the object, and have self-similarity between parts and overall features of the object. The smaller-scale structure of fractals are similar to the larger-scale structure. Hence, fractals do not have a single characteristic scale. Many anatomical objects, such as cortical surfaces and the cardiovascular system, are self-similar. The complexity of such objects can be quantified using the fractal dimension (FD). The main question is if brain networks exhibit the characteristic of self-similarity. To answer this question, we need to compute the FD.

Let ϵ be the scale and N_ϵ be the number of self-similar parts that can cover the whole structure. Then FD is defined as

$$\mathrm{FD} = \lim_{\epsilon \to 0} \frac{\ln N_\epsilon}{\ln \frac{1}{\epsilon}}.$$

In practice, we cannot compute the limit as ϵ goes to zero for real anatomical objects, so we resort to the box-counting method (Hutchinson, 1981; Mandelbrot, 1982). For k different scales $\epsilon_1, \ldots, \epsilon_k$, we have a corresponding number of covers $N_{\epsilon_1}, \ldots, N_{\epsilon_k}$. Then we draw the log–log plot of $(N_{\epsilon_i}, 1/\epsilon_i)$ and fit a linear line in a least squares fashion. The slope of the fitted line is the estimated FD. We can also estimate the FD locally using finite differences with neighboring measurements as

$$FD = -\frac{\Delta \ln N_\epsilon}{\Delta \ln \epsilon},$$

where Δ is the second-order finite difference (Figure 3.6).

For networks, we avoided using the box-counting method to the space where the network is embedded and the graph itself (Song et al., 2005). Rather, we have applied the method to the structure that defines link connections, i.e., the adjacency matrix. The use of the adjacency matrix simplifies a lot of computation. The Erdös–Rényi random graph $G(n, p)$ is defined as a random graph generated with n nodes where two nodes are connected with probability p (Figure 3.6) (Erdös and Rényi, 1961). The FD characteristic of the brain networks is different from that of random graphs.

Other graph theory features. Although we do not discuss them in detail here, other various popular graph theoretic measures are also proposed: entropy (Sporns et al., 2000), hub centrality (Freeman, 1977), and modularity. A review of various graph measures can be found in Bullmore and Sporns (2009). The centrality of a node measures the number of shortest paths between

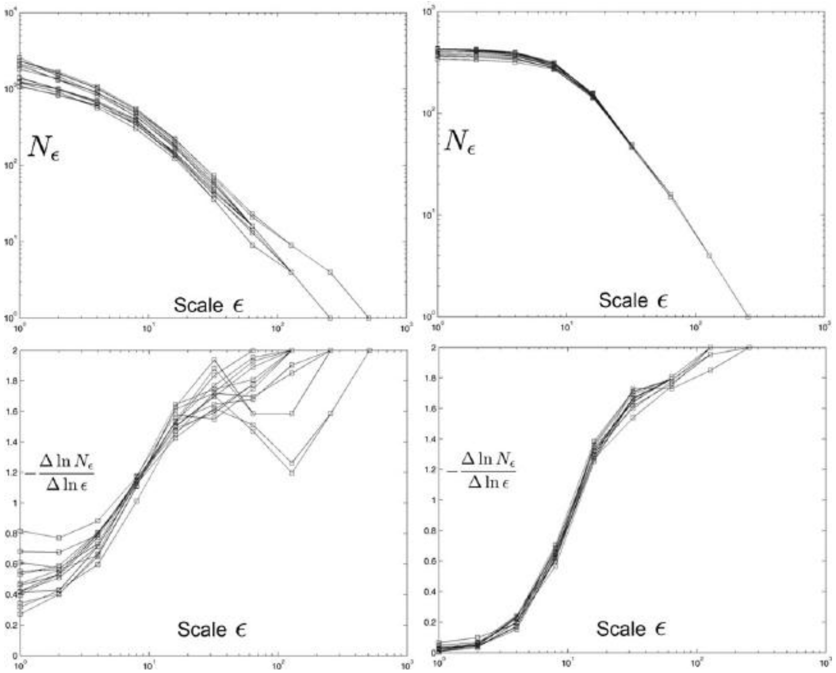


Figure 3.6 The plots of $\ln N_\epsilon$ and $-\frac{\Delta \ln N_\epsilon}{\Delta \ln \epsilon}$ over scale $\ln \epsilon$. The first column is for 14 normal control subjects and the second column is for 14 Erdös–Rényi random graphs $G(200, 0.01)$. The FD characteristic of the random graph is different from the brain networks.

all other nodes that pass through the given node, and is motivated in part by modeling the social network (Freeman, 1977). Nodes with high centrality, which are likely to be with high degree, are called hubs. The modularity of a network measures the number of components or modules and is related to hierarchical clustering (Girvan and Newman, 2002; Lee et al., 2012).

4
Correlation Networks

Correlation-based networks are probably the most often used network model in brain imaging. In this chapter, we study Pearson's product-moment correlation (Fisher, 1915), in short *Pearson correlation*. In sciences, Pearson correlation has been widely used as a simple index for measuring dependency and the linear relationship between two variables. In human brain mapping research, it has been mainly used to map out functional or anatomical connectivity (Friston et al., 1993a; Worsley et al., 2005b; Chung et al., 2015a).

4.1 Pearson Correlations

Definition 4.1 *Consider two data vectors* $\mathbf{x} = (x_1, x_2, \cdots, x_n)^\top$ *and* $\mathbf{y} = (y_1, y_2, \cdots, y_n)^\top$. *The Pearson* correlation *coefficient* ρ *between two vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *is defined as*

$$\rho(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \tag{4.1}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i$ *and* $\bar{y} = \frac{1}{n}\sum_{i=1}^n y_i$ *are the sample means.*

Then algebraic manipulation can show that

$$\rho(\mathbf{x},\mathbf{y}) = \rho(a\mathbf{x} + b, c\mathbf{y} + d) \tag{4.2}$$

for any nonzero $a, c \in \mathbb{R}$ and any $b, d \in \mathbb{R}$. Thus, the correlation is scale and translation invariant. The correlation (4.1) can be factored as a vector product:

$$\rho(\mathbf{x},\mathbf{y}) = [\mathbf{x}']^\top \mathbf{y}', \tag{4.3}$$

where $\mathbf{x}' = \alpha\mathbf{x} + \beta$ and $\mathbf{y}' = \gamma\mathbf{y} + \delta$ such that

$$\alpha = \frac{1}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}}, \quad \beta = -\frac{\bar{x}}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}},$$

$$\gamma = \frac{1}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}, \quad \delta = -\frac{\bar{y}}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}.$$

Then trivially, correlations under *any* nontrivial linear transformation, i.e., $a, c \neq 0$, can be represented as vector products due to the invariance (4.2).

Among all linear transformations, following linear transformation is the most often used in relation to sparse models (Chung et al., 2015a, 2017a). Consider a transformation that center and scale $\mathbf{x}$ and $\mathbf{y}$ such that

$$\sum_{i=1}^n x_i = \sum_{i=1}^n y_i = 0,$$

$$\|\mathbf{x}\|^2 = \mathbf{x}^\top\mathbf{x} = \|\mathbf{y}\|^2 = \mathbf{y}^\top\mathbf{y} = 1. \tag{4.4}$$

This projects n-dimensional vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ onto a n-dimensional unit sphere S^{n-1}. Thus the centering and scaling operations can be viewed as data embedding from higher Euclidean space $\mathbb{R}^n$ onto unit sphere S^{n-1}.

Then the correlation between $\mathbf{x}$ and $\mathbf{y}$ is simply the *cosine similarity* often used in engineering and computer science literature:

$$\cos\theta = \mathbf{x}^\top\mathbf{y},$$

where θ is the angle between vectors $\mathbf{x}$ and $\mathbf{y}$.

Assuming $\mathbf{x}$ and $\mathbf{y}$ are centered and scaled, consider the following linear model

$$\mathbf{y} = \beta\mathbf{x} + \mathbf{e},$$

where $\mathbf{e}$ is a mean zero error vector and β is the unknown scalar parameter we need to estimate. The least squares estimation (LSE) of β is given by minimizing the sum of the squared residuals:

$$\mathbf{e}^\top\mathbf{e} = (\mathbf{y} - \beta\mathbf{x})^\top(\mathbf{y} - \beta\mathbf{x}).$$

Trivially we can show that LSE of β is

$$\widehat{\beta} = \mathbf{x}^\top\mathbf{y},$$

the Pearson correlation. Even if $\mathbf{x}$ and $\mathbf{y}$ are not centered and scaled, there is a relationship between correlations and the regression coefficients.

4.2 Partial Correlations

Let $Y = (Y_1, Y_2)$ be two variables of interests and $X = (X_1, \cdots, X_p)$ be a row vector of variables that should be removed in a data analysis. For instance, we may let Y_1 and Y_2 be functional activity at two different voxels, and X_1 and X_2 be the age and gender. The covariance matrix of $(Y, X)^\top$ is denoted by

$$\mathbb{V}(Y,X)^\top = \begin{pmatrix} \Sigma_{YY} & \Sigma_{YX} \\ \Sigma_{XY} & \Sigma_{XX} \end{pmatrix}. \tag{4.5}$$

Note Σ_{XY} is the cross-covariance matrix of X and Y. Σ_{YX}, Σ_{XX}, and Σ_{YY} are defined similarly. Then the partial covariance of Y given X is

$$(\sigma_{ij}) = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}.$$

The *partial correlation* $\rho_{Y_i,Y_j|X}$ is the correlation between variables Y_i and Y_j while removing the effect of variables X, and it is defined as

$$\rho_{Y_i,Y_j|X} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}.$$

The *conditional* notation | is used in defining the partial correlation since the partial correlation is equivalent to *conditional correlation* if

$$\mathbb{E}(Y|X) = a + BX$$

for some vector a and matrix B, which is true under the normality of data. This is the formulation we used to compute the partial correlation. If vector X consists of a single measurement, i.e., $X = X_1$, the partial correlation can be computed from the simple correlation via

$$\rho_{Y_1,Y_2|X} = \frac{\rho_{Y_1,Y_2} - \rho_{Y_1,X}\rho_{Y_2,X}}{\sqrt{(1-\rho_{Y_1,X}^2)(1-\rho_{Y_2,X}^2)}}.$$

The preceding definition assumes $\rho_{Y_1,X}$ and $\rho_{Y_2,X}$ are not exactly -1 or 1. The *sample partial correlation* $r_{Y_1,Y_2|x}$ is defined similarly by replacing the covariance with the sample covariance in (4.5).

The MATLAB codes for computing the partial correlation are as follows. Let `rho` be the sample partial correlation between time series fMRI1 and fMRI2 while removing the effect of age (`age`) and gender (`gender`) effects. For n subjects in the group, all variables are row vectors of size $1 \times n$. Then we compute the partial correlation `rho` as

```
x=[age; gender];
y=[fMRI1; fMRI2];
a=cov([x;y]');
```

```
b=a(1:2,1:2)-a(1:2,3:4)*inv(a(3:4,3:4))*a(3:4,1:2);
rho=b(1,2)/sqrt(b(1,1)*b(2,2));
```

Here x and y are $2 \times n$ matrices, and the covariance matrix a is the size 4×4.

4.3 Averaging Correlations

In brain imaging, it frequently happens that the addition is not a well-defined concept. Correlation is such an example. Given two correlations r_1 and r_2, it is possible to have $r_1 + r_2 < 1$ or $r_1 + r_2 > 1$. Thus, the sum of correlations may not be correlation. Subsequently, the average of correlation may not be a well-defined concept. In this section, we show how to properly average correlations and correlation matrices. We start with vector space.

4.3.1 Vector Spaces

Definition 4.2 *A vector space $\mathcal{S}$ is a collection of objects satisfying various axioms of algebraic rules. Given $x, y, z \in \mathcal{S}$, we need to have*

$$\begin{aligned} x + y = y + x &\in \mathcal{S} \ \ \textit{(commutative)},\\ x + (y + z) = (x + y) + z &\in \mathcal{S} \ \ \textit{(associative)}. \end{aligned}$$

It also requires us to have identity $0 \in \mathcal{S}$ such that

$$0 + x = x$$

and inverse $-x \in \mathcal{S}$ such that

$$x + (-x) = 0.$$

Given $a, b \in F$, a field,

$$\begin{aligned} a(bx) &= (ab)x \ \ \textit{(compatibility)},\\ 1x &= x \ \ \textit{for some } 1 \in F \ \textit{(identity)}\\ a(x + y) &= ax + ay \ \ \textit{(distributivity w.r.t. vector addition)}\\ (a + b)x &= ax + bx \ \ \textit{(distributivity w.r.t. field addition)}. \end{aligned}$$

Given any $x, y \in \mathcal{S}$ and $a, b \in \mathbb{R}$, if $ax + by \in \mathcal{S}$, $\mathcal{S}$ is most likely a vector space in practice. Given two correlations r_1 and r_2, it is possible to have $r_1 + r_2 < -1$ or $r_1 + r_2 > 1$. Thus, correlations do not form a vector space.

Definition 4.3 *Given objects $f_1, \cdots, f_n \in \mathcal{S}$ for some space S, the (sample) mean $\bar{f}$ of the objects is defined as*

$$\bar{f} = \frac{1}{n}\sum_{j=1}^{n} f_j. \tag{4.6}$$

Thus, the sample mean is only defined in a vector space. A space that is not a vector space may not have the properly defined sample mean. To properly do a statistical inference, it is a necessity to have a vector space at least.

In the Euclidean space $\mathcal{S}$, the sample mean is the minimizer of the following cost function.

$$\bar{f} = \arg\min_{g \in \mathcal{S}} \sum_{j=1}^{n} \|g - f_j\|_2^2,$$

where $\|\cdot\|_2$ is the L_2-norm. This is easily proved by noting that the cost function is quadratic in norm $\|g\|_2$. By differentiating the cost function with respect to $\|g\|_2$ and setting the differentiation equal to zero, we obtain the minimum.

4.3.2 Averaging by Back Projection

The average $\bar{f}$ may not belong to $\mathcal{S}$ if $\mathcal{S}$ is not a vector space. Thus, to define average, $\mathcal{S}$ must be a vector space. If $\mathcal{S}$ is not a vector space, we transform $\mathcal{S}$ to vector space $\mathcal{T}$ using some nonlinear transform

$$\mathcal{F} : \mathcal{S} \to \mathcal{T}.$$

We assume the inverse of $\mathcal{F}$ is well defined and easy to compute. $\mathcal{F}$ has to be one-to-one to make any sense in the following operations.

Given any $f_1, \cdots, f_n \in \mathcal{S}$, we have

$$g_j = \mathcal{F}(f_j) \in \mathcal{S}.$$

Then the average in $\mathcal{T}$ is defined as

$$\bar{g} = \frac{1}{n}\sum_{j=1}^{n} g_j = \frac{1}{n}\sum_{j=1}^{n} \mathcal{F}(f_i).$$

This new point $\bar{g}$ can be back projected into $\mathcal{F}$ via $\mathcal{F}^{-1}(\bar{g})$. Thus the reasonable average in $\mathcal{S}$ is defined as

$$\bar{f} = \mathcal{F}^{-1}\Big[\frac{1}{n}\sum_{j=1}^{n} \mathcal{F}(f_j)\Big].$$

The back projection method can be used to average correlations. The sample mean of correlations is not a well-defined concept. Over the years, a number of different methods for averaging correlations have been proposed. Silver and Hollingsworth (1989) proposed to transform correlations into z-scores by Fisher's transform (Fisher, 1915):

$$F(\rho) = \text{arctanh}(\rho) = \frac{1}{2}\ln\frac{1+\rho}{1-\rho}.$$

Note the inverse of the Fisher's transform is given by

$$F^{-1}(z) = \frac{e^{2z}-1}{e^{2z}+1}.$$

Given a collection of correlations $\rho_1, \rho_2, \cdots, \rho_n$, the average of Fisher's transforms is

$$z = \frac{1}{n}\sum_{i=1}^{n} F(\rho_i).$$

Then $\mathcal{F}^{-1}(z)$ is the average of correlations via the back projection. This approach is known to introduce bias (Corey et al., 1998). Another slightly less obvious approach is to use the log-Euclidean distance.

4.3.3 Metric Spaces

Diffusion tensor images can produce 3D unit vectors that measure the principle direction of diffusion of water molecules at each voxel. Given n unit vectors $\mathbf{v}_1, \cdots, \mathbf{v}_n$ with $\|\mathbf{v}_j\| = 1$, the average diffusion direction is not well defined in the following sense. Consider collection $S^2$of all the unit vectors. Obvious, $\mathbf{v}_1, \cdots, \mathbf{v}_n \in S^2$. However, $\bar{\mathbf{v}} = \sum_{j=1}^{n} \mathbf{v}_j/n \notin S^2$. $\mathbf{v}_1, \cdots, \mathbf{v}_n$ are points along the 3D unit sphere. $\bar{\mathbf{v}}$ is inside the 3D unit solid ball. To properly define averaging operation in this example, we need a concept called the Fréchet mean, which is defined in metric spaces. The metric spaces were introduced by Fréchet in 1906 as a part of his PhD thesis. Fréchet axiomatized the notion of distance and showed that many abstract spaces are metric spaces simplifying various difficult problems to that of metric properties.

Definition 4.4 *A* metric space *is a collection of objects for which the pairwise distance between objects is well defined. The distances are called* metric. *Formally,* $(\mathcal{M}, d)$ *is a metric space if* d *satisfies the condition*

$$d(x,y) = 0 \leftrightarrow x = y. \textit{ (identity)}$$

$$d(x,y) = d(y,x) \textit{ (symmetry)}$$

$$d(x,z) \le d(x,y) + d(y,z) \textit{ (triangle inequality)}$$

Then from the three axioms of metric, it can be shown that metric is always nonnegative, i.e.,

$$d(x,y) \geq 0.$$

This is left as an exercise.

There are numerous metric spaces. The usual Euclidean space $\mathbb{R}^n$ is a metric space with L_2-norm

$$d(x,y) = \|x-y\|_2 = \left[\sum_{i=1}^{n}(x_i - y_i)^2\right]^{1/2},$$

where $x = (x_1, \cdots, x_n)^\top$ and $y = (y_1, \cdots, y_n)^\top$.

In another example, let $\mathcal{M}$ be a collection of $n \times n$ matrices that represent a graph with n nodes. Given $W^1 = (w_{ij}^1)$, $W^2 = (w_{ij}^2) \in \mathcal{M}$, define L_l-distance as

$$D_l(W^1, W^2) = \parallel w^1 - w^2 \parallel_l = \left(\sum_{i,j} \left|w_{ij}^1 - w_{ij}^2\right|^l\right)^{1/l}.$$

When $l = \infty$, L_∞-distance is written as

$$D_\infty(W^1, W^2) = \parallel w^1 - w^2 \parallel_\infty = \max_{\forall i,j} \left|w_{ij}^1 - w_{ij}^2\right|.$$

Then we can show that $(\mathcal{M}, D_l)$ and $(\mathcal{M}, D_\infty)$ are metric spaces. This is left as an exercise.

The elementwise matrix distances differences may not necessarily be the best distance for matrices. L_1 and L_2-distances usually suffer the problem of outliers. Few outlying extreme edge weights may severely affect the distance. Further, these distances ignore the underlying topological structures. There exists a more topologically sensitive network distances (Chung et al., 2017a,d), which we will study later.

4.3.4 Fréchet Mean

Generalizing the idea in (4.7), we define the Fréchet mean of $f_1, \cdots, f_n \in \mathcal{M}$ with metric d as

$$\bar{f} = \arg\min_{f \in \mathcal{M}} \sum_{j=1}^{n} d(f, f_j)^2.$$

In the Euclidean space, the sample mean and the Fréchet mean are identical. If $\mathcal{M}$ is not a vector space, the Fréchet mean may not be the sample mean.

On the sphere, the shortest distance between any two points $\mathbf{v}$ and $\mathbf{v}_j$ is the shortest arc in of the greatest circle passing through the two points. The arc length is the angle

$$\theta_j = \cos^{-1}\left(\mathbf{v}_j^\top \mathbf{v}\right).$$

We can show that the arc length is a metric. Then, the Fréchet mean of diffusion direction is given by

$$\bar{\theta} = \min_{\mathbf{v} \in S^2} \sum_{j=1}^{n} \left[\cos^{-1}\left(\mathbf{v}_j^\top \mathbf{v}\right)\right]^2.$$

The numerical implementation of computing the Fréchet mean on a sphere is left as an exercise. This is not a trivial problem on the sphere since it is mathematically not possible to have uniform grids on the sphere. Also, the Fréchet mean may not be unique for some pathological example on S^2.

The Fréchet mean can be used to average the covariance and correlation matrices for brain network modeling. Qiu et al. (2015) is the first paper that averaged the collection of brain networks using the Fréchet mean. This requires defining a metric in the space of symmetric positive definite matrices, which is not a trivial problem. The metric is given by the log-Euclidean distance (Arsigny et al., 2005, 2006, 2007), which we will study later. The log-Euclidean framework is often used in averaging diffusion tensor images.

4.3.5 Log-Euclidean Distance

The concept of Fréchet mean can be used to average correlation or covariance matrices.

Definition 4.5 *Given symmetric and positive definite matrix $C = (c_{ij})$, its* matrix exponential *is defined as a Taylor expansion:*

$$e^C = I + C + \frac{1}{2!}C^2 + \frac{1}{3!}C^3 + \cdots,$$

where I is the identity matrix. If there exists matrix A satisfying $e^A = C$, then A is called the matrix logrithm *of C and denoted as* $\log C$.

For symmetric positive definite $p \times p$ matrix C, the logarithm of C can be computed as follows. Note C has p positive eigenvalues $\lambda_1, \cdots, \lambda_p$. There exists an orthogonal matrix Q such that

$$C = Q^\top D Q$$

with $D = diag(\lambda_1, \cdots, \lambda_p)$, the diagonal matrix consisting of entries $\lambda_1, \cdots, \lambda_p$. Then using the fact $(Q^\top DQ)^k = Q^\top D^k Q$,

$$e^C = I + Q^\top DQ + \frac{1}{2!}Q^\top D^2 Q + \frac{1}{3!}Q^\top D^3 Q + \cdots = Q^\top e^D Q.$$

Thus the exponential of C is $Q^\top e^D Q$. Similarly, the logarithm of C is $Q^\top \log DQ$.

If matrix C is nonnegative definite with zero eigenvalues, the matrix logarithm is *not* defined since $\log 0$ is not defined. Thus, we cannot apply logarithm directly to rank-deficient large correlation and covariance matrices obtained from small number of samples. One way of applying logarithm to nonnegative definite matrices is to make matrix C diagonally dominant by adding a diagonal matrix αI with suitable choice of relatively large α (Chan and Wood, 1997).

Definition 4.6 *Given two symmetric and positive definite matrices C_1 and C_2, the* log-Euclidean *distance between C_1 and C_2 is given by*

$$d(C_1, C_2) = \| \log C_1 - \log C_2 \|_F,$$

where the Frobenius norm is defined as (Arsigny et al., 2005, 2006, 2007)

$$\|A\|_F = \sqrt{tr(A^\top A)}.$$

The log-Euclidean distance can be written differently as

$$d(C_1, C_2) = \Big[tr\,(\log C_1 - \log C_2)^2\Big]^{1/2}.$$

The log-Euclidean distance can be viewed as the generalized manifold version of Frobenius norm distance. Given a collection of correlation matrices $C_1, C_2, \cdots, C_n$, *log-Euclidean Fréchet mean* $\bar{C}$ is given by (Arsigny et al., 2007)

$$\bar{C} = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \log C_i\right). \tag{4.7}$$

Since correlation is the off-diagonal entries of correlation matrices, the diagonal entries are the average of correlations in the log-Euclidean sense.

Limitation of log-Euclidean distance. If a matrix is nonnegative definite with zero eigenvalues, the matrix logarithm is *not* defined since $\log 0$ is not defined. Thus, we cannot apply the logarithm directly to rank-deficient large correlation and covariance matrices obtained from data with small sample sizes relative to the number of nodes. One way of applying the logarithm to nonnegative definite matrices is to make the matrix diagonally dominant by adding a diagonal matrix αI with suitable choice of relatively large α

(Chan and Wood, 1997). Alternately, we can perform a graphical LASSO-type of sparse model and obtain the closest positive definite matrices (Qiu et al., 2015; Mazumder and Hastie, 2012). Developing the log-Euclidean for general correlation matrices including nonnegative ones is beyond the scope of this chapter. Note topological distances are applicable to nonnegative definite connectivity matrices.

4.4 Correlation as Metric

Consider a node set $V = \{1, \ldots, p\}$ and edge weights $\rho = (\rho_{ij})$, where ρ_{ij} is the weight between nodes i and j. The edge weights measure similarity or dissimilarity between nodes. The edge weights in most brain networks are usually given by some similarity measure between nodes (Mclntosh and Gonzalez-Lima, 1994; Newman and Watts, 1999; Song et al., 2005; Li et al., 2009; Lee et al., 2011a). Weighted network $X = (V, \rho)$ is formed by the pair of node set V and edge weights ρ. If X is a metric, network interpretation is straightforward.

We will show how to construct a metric using correlations.

Consider $n \times 1$ measurement vector $\mathbf{x}_j = (x_{1j}, \cdots, x_{nj})^\top$ on node j. Suppose we center and rescale the measurement $\mathbf{x}_j$ such that

$$\| \mathbf{x}_j \|^2 = \mathbf{x}_j'\mathbf{x}_j = \sum_{i=1}^{n} x_{ij}^2 = 1$$

and

$$\sum_{i=1}^{n} x_{ij} = 0.$$

Naturally, we are interested in using correlations or their simple functions as edge weights, i.e.,

$$\rho_{ij} = \mathbf{x}_i^\top \mathbf{x}_j \quad \text{or} \quad \rho_{ij} = 1 - \mathbf{x}_i^\top \mathbf{x}_j.$$

However, not every function of correlations is metric.

Example 4.1 [1] *$\rho_{ij} = 1 - \mathbf{x}_i^\top \mathbf{x}_j$ is* not *a metric. Consider the following three-node counterexample:*

$$\mathbf{x}_i = \left(0, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)^\top,$$

[1] The counterexample is provided by Zhiwei Ma of University of Chicago.

$$\mathbf{x}_j = \left(\frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}}\right)^\top,$$
$$\mathbf{x}_k = \left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, -\frac{2}{\sqrt{6}}\right)^\top.$$

Then we have $\rho_{ij} > \rho_{ik} + \rho_{jk}$.

Then the interesting methodological question is to identify minimum conditions that make a function of correlations a metric.

Theorem 4.1 *For centered and scaled data* $\mathbf{x}_1, \cdots, \mathbf{x}_p$*, let*

$$\rho_{ij} = cos^{-1}(\mathbf{x}_i^\top \mathbf{x}_j).$$

Then ρ_{ij} *is metric.*

Proof. On unit sphere S^{n-1}, the correlation between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the cosine angle θ_{ij} between the two vectors, i.e.,

$$\mathbf{x}_i^\top \mathbf{x}_j = \cos\theta_{ij}.$$

The geodesic distance ρ between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ on the unit sphere is given by angle θ_{ij}:

$$\rho(\mathbf{x}_i, \mathbf{x}_j) = \cos^{-1}(\mathbf{x}_i^\top \mathbf{x}_j).$$

For nodes $\mathbf{x}_i, \mathbf{x}_j \in S^{n-1}$, there are two possible angles θ_{ij} and $2\pi - \theta_{ij}$ depending on if we measure the angles along the shortest arc or longest arc. We take the convention of using the smallest angle in defining θ_{ij}. With this convention,

$$\rho(\mathbf{x}_i, \mathbf{x}_j) \leq \pi.$$

Given three nodes $\mathbf{x}_i, \mathbf{x}_j$, and $\mathbf{x}_k$, which forms a spherical triangle, we then have spherical triangle inequality

$$\rho(\mathbf{x}_i, \mathbf{x}_j) \leq \rho(\mathbf{x}_i, \mathbf{x}_k) + \rho(\mathbf{x}_k, \mathbf{x}_j).$$

The proof to (4.8) is given in Reid and Szendròi (2005). Thus we proved ρ is a metric. □

Theorem 4.2 *For any metric* ρ_{ij}*,* $f(\rho_{ij})$ *is also a metric if* $f(0) = 0$ *and* $f(x)$ *is increasing and concave for* $x > 0$.

The proof is given in Van Dijk et al. (2012). Such function f is called the *metric preserving function.*

Any power $[cos^{-1}(\mathbf{x}_i^\top \mathbf{x}_j)]^{1/m}$ for $m \geq 1$ is metric. When $m = 1$, we have the simplest possible metric $\rho(\mathbf{x}_i, \mathbf{x}_j) = \cos^{-1}(\mathbf{x}_i^\top \mathbf{x}_j)$, which obtains minimum 0 when $\mathbf{x}_i^\top \mathbf{x}_j = 1$ and maximum π when $\mathbf{x}_i^\top \mathbf{x}_j = -1$.

Theorem 4.3 *For any* $\mathbf{x}_1, \cdots, \mathbf{x}_p \in \mathbb{R}^n$,

$$\rho_{ij} = \left[1 - corr(\mathbf{x}_i, \mathbf{y}_j)\right]^{1/2}$$

is a metric, where $corr(\mathbf{x}_i, \mathbf{y}_j)$ *is the Pearson correlation.*

4.5 Statistical Inference on Correlations

Regardless of if we have Pearson correlations or partial correlations, the statistical inference can be done similarly.

4.5.1 Inference on One Sample

Let $\rho(p)$ be correlation or partial correlation for each voxel p; we are interested in testing

$$H_0 : \rho(p) = \rho \quad \text{vs.} \quad H_1 : \rho(p) \neq \rho \tag{4.8}$$

for some fixed ρ. In the usual one-sample inference, we simply test if correlation $\rho(p)$ each voxel is 0 or not. Inference type (4.8) is useful if only one sample is available or determining high-correlation regions within the brain. There are many different ways for testing the preceding hypotheses. One widely used technique is to use the Fisher transform (Fisher, 1915) that transforms the sample correlation r into

$$F(r) = \text{arctanh}(r) = \frac{1}{2} \ln \frac{1+r}{1-r}.$$

Then for moderately large samples, $F(r)$ shows asymptotic normality:

$$F(r) \sim N\Big(\frac{1}{2} \ln \Big(\frac{1+\rho_k}{1-\rho_k}\Big), \frac{1}{n_k - 3}\Big).$$

The transform can be viewed as a variance-stabilizing normalization process. Then the test statistic under null is

$$Z = \sqrt{n-3}[F(r) - F(\rho)] \sim N(0,1),$$

which is a standard normal distribution (Chung, 2007) (see Figure 4.1).

Another way of testing the significance is to use the t-statistic. Assuming the normality of data, the sample correlation or partial correlation r can be transformed to be distributed as follows:

$$T = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \sim t_{n-2},$$

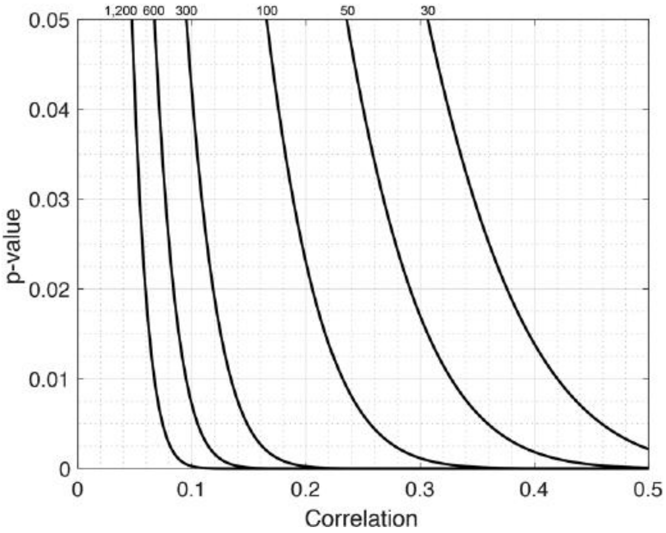


Figure 4.1 p-value over correlation in one sample. The numbers at the top are sample sizes. For a large sample size of 1,200, even correlation of 0.1 gives the statistical significance below 0.01. However, for a small sample size of 30, correlation 0.5 does not give the statistical significance 0.01.

the t distribution with $n - 2$ degrees of freedom. This test statistic can be used for testing hypothesis (4.8).

Here n is the sample size. If we are correlating time series with n time points, the number of time points is the sample size.

4.5.2 Inference on Two Samples

Let $\rho_1(p)$ and $\rho_2(p)$ be two independent correlations at position p. We are interested in testing the equality of correlations. At each fixed point p, we are interested in testing

$$H_0 : \rho_1(p) = \rho_2(p) \text{ vs. } H_1 : \rho_1(p) \neq \rho_2(p). \tag{4.9}$$

For two sample inference type (4.9), the test statistic under H_0 is given by the following:

$$W(p) = \frac{\ln\left(\frac{1+r_1}{1-r_1} \cdot \frac{1-r_2}{1+r_2}\right)}{2\sqrt{\frac{1}{n_1-3} + \frac{1}{n_2-3}}} \sim N(0,1), \tag{4.10}$$

where r_1 and r_2 are the sample means that estimate $\rho_1(p)$ and $\rho_2(p)$ at fixed point p.

4.6 Cosine Series Representation

EEG at a channel and fMRI at a voxel are time series. Many time series analysis techniques have been applied in correlating two functional signals at different spatial locations. Since functional signals are highly noise, often a Fourier transform type of signal filtering is applied before the transform. Here we briefly explain the cosine series representation, which is the analytic version of the often used cosine Fourier transform (Chung et al., 2010a). The cosine series representation can be viewed as a type of Fourier descriptor. Fourier descriptors have been around for many decades for modeling noise functional data and planar curves (Persoon and Fu, 1977; Staib and Duncan, 1992). They have been previously used to classify curves (Batchelor et al., 2006), where the Fourier coefficients are computed by the Fourier transform that involves both the sine and cosine series expansion. Then the sum of the squared coefficients are obtained up to a certain degree for each functional data and the k-means clustering is used to classify the data. The cosine series representation differs from (Batchelor et al., 2006) in that we represent functional data employing cosine series only, without using both the cosine and sine series making the representation more compact. Matlab implementation for the cosine series representation is available.[2]

4.6.1 Eigenfunctions of Laplacian in a Unit Interval

There are infinitely many possible orthonormal bases in interval $[0,1]$. Here we explain the spectral approach for obtaining orthonormal bases.

Consider the space of square integrable functions in $[0,1]$ denoted by $\mathcal{L}^2[0,1]$. Let us solve the eigenequation

$$\Delta\psi + \lambda\psi = 0$$

in $\mathcal{L}^2[0,1]$ with 1D Laplacian $\Delta = \frac{d^2}{dt^2}$. Then it can be shown that the eigenfunctions $\psi_0, \psi_1, \cdots$ form an orthonormal basis in $\mathcal{L}^2[0,1]$. Note that if ψ_j is an eigenfunction, any multiple of ψ_j is also eigenfunction. Thus, it is expected the eigenfunctions are properly normalized. The eigenfunctions satisfying (8.2) are then given by the usual Fourier sine and cosine basis

$$\psi_0(t) = 1, \psi_l = \sqrt{2}\sin(l\pi t), \sqrt{2}\cos(l\pi t)$$

with the corresponding eigenvalues $\lambda_l = l^2\pi^2$.

Note that there are two eigenfunctions corresponding to the same eigenvalue. Note that the multiplicity of eigenfunctions only happens if there is

[2] http://brainimaging.waisman.wisc.edu/~chung/tracts/

a symmetry in the domain of the eigenvalue problem. The constant $\sqrt{2}$ is introduced to make the eigenfunctions orthonormal in $[0,1]$ with respect to the inner product

$$\langle f, g\rangle = \int_0^1 f(t)g(t)\,dt. \tag{4.11}$$

With respect to the inner product, the norm $\|\cdot\|$ is then defined as

$$\|f\| = \langle f, f\rangle^{1/2}.$$

Using both sine and cosine bases is not algebraically efficient. Instead of solving (8.2) in the domain $[0,1]$, consider solving the problem in the larger unbounded domain $\mathbb{R}$ with the periodic constraint

$$\psi(t+2) = \psi(t).$$

The period 2 constraint forces the basis function expansion to be only valid in the intervals $\cdots, [-2, -1], [0,1], [2,3], \cdots$ while there are gaps in $\cdots, (-1,0), (1,2), (3,4), \cdots$. We can fill the gap by padding with some arbitrary function. However, if we pad the gaps with any function, it may result in the Gibbs phenomenon (ringing artifacts) at the boundary of the intervals $\cdots, 2, 1, 0, 1, 2, \cdots$ (Chung et al., 2007). To avoid the Gibbs phenomenon, we force the function to be continuous at the boundary by putting the constraint of evenness, i.e.,

$$\psi(t) = \psi(-t).$$

If $\psi(t)$ is the eigenfunction well defined in $[0,1]$, in the intervals, $\cdots, [-2, -1], (-1,0)[0,1], (1,2), [2,3], \cdots$ we must have

$$\cdots, \psi(t-2), \psi(-t), \psi(t), \psi(-t+2), \psi(t+2), \cdots.$$

The only eigenfunctions satisfying the two constraints (4.12) and (4.12) are the cosine basis

$$\psi_0(t) = 1, \psi_l(t) = \sqrt{2}\cos(l\pi t) \tag{4.12}$$

with the corresponding eigenvalues $\lambda_l = l^2\pi^2$ for integers $l > 0$. Then using the cosine basis only, any $f \in \mathcal{L}^2[0,1]$ can be represented as

$$f(t) = \sum_{l=0}^{k} c_l \psi_l(t) + \epsilon(t),$$

where c_l are the Fourier coefficients and ϵ is the residual error for using only kth degree expansion.

Note that it is possible to put the constraint of oddness, i.e.,

$$\psi(t) = -\psi(-t).$$

Then we have sine basis

$$\psi_l(t) = \sqrt{2}\sin(l\pi t). \tag{4.13}$$

4.6.2 Fourier Series

Consider ith functional time series data

$$\zeta_i(t) = \mu_i(t) + \epsilon_i(t),$$

where t is the time variable. We assume the functional data are scaled in such a way that they are defined in $[0, 1]$. Formulating the Fourier analysis in a unit interval makes the numerical implementation more convenient. ϵ_i is a zero mean noise at each fixed t, i.e.,

$$\mathbb{E}\epsilon_i(t) = 0.$$

μ_i is an unknown smooth function to be estimated. It is reasonable to assume that

$$\zeta_i, \mu_i \in \mathcal{L}^2[0, 1],$$

the space of square integrable functions. Any function $f \in \mathcal{L}^2[0, 1]$ satisfies the condition

$$\int_0^1 f^2(t)\, dt < \infty.$$

This condition is needed to guarantee the convergence in the Fourier series.

Instead of estimating μ_i in $\mathcal{L}^2[0, 1]$, we estimate it in a smaller subspace $\mathcal{H}_k$, which is spanned by up to the k orthonormal basis functions:

$$\mathcal{H}_k = \left\{ \sum_{l=0}^{k} c_l \psi_l(t) : c_l \in \mathbb{R} \right\} \subset \mathcal{L}^2[0, 1].$$

Then the LSE of μ_i in $\mathcal{H}_k$ is given by

$$\widehat{\mu_i} = \arg \min_{f \in \mathcal{H}_k} \left\| f - \zeta_i(t) \right\|^2. \tag{4.14}$$

Theorem 4.4 *The minimization of* (4.14) *is given by*

$$\widehat{\mu_i} = \sum_{l=0}^{k} \langle \zeta_i, \psi_l \rangle \psi_l,$$

where the lth degree Fourier coefficient $\langle \zeta_i, \psi_l \rangle$ *is given by the inner product*

$$\langle \zeta_i, \psi_l \rangle = \int_0^1 \zeta_i(t)\psi_l(t)\, dt. \tag{4.15}$$

Proof. Heuristically, we need to find function

$$f(t) = \sum_{l=0}^{k} c_l \psi_l(t)$$

that is the closest to ζ_i. The distance between f and ζ_i is given by

$$I(c_0, c_1, \cdots, c_k) = \int_0^1 \Big| \sum_{l=0}^{k} c_l \psi_l(t) - \zeta_i(t) \Big|^2 dt,$$

which is a $k+1$ dimensional function in unknown parameter space $(c_0, c_1, \cdots, c_k) \in \mathbb{R}^{k+1}$. Since I is a quadratic function in $(c_0, c_1, \cdots, c_k)$, it has the global minimum at

$$\frac{\partial I}{\partial c_0} = \frac{\partial I}{\partial c_1} = \cdots = \frac{\partial I}{\partial c_k} = 0.$$

The algebraic derivation is left as an exercise, but we have

$$c_l = \langle \zeta_i, \psi_l \rangle$$

for all $l = 0, 1, \cdots, k$. □

The expansion (4.15) is called the kth degree *Fourier series*. As $k \to \infty$, the expansion converges to ζ_i, i.e.,

$$\zeta_i(t) = \sum_{l=0}^{\infty} \langle \zeta_i, \psi_l \rangle \psi_l.$$

It is also possible to have a slightly different but equivalent model that is easier to use in statistical inference. Assuming Gaussianness of data, $\epsilon_i(t)$ is a Gaussian stochastic process, which is simply a collection of random variables. Then $\epsilon_i(t)$ can be expanded using the given basis ψ_l as follows:

$$\epsilon_i(t) = \sum_{l=0}^{k} Z_l \psi_l(t) + e_i(t),$$

where $Z_l \sim N(0, \tau_l^2)$ are possibly *correlated* Gaussian random variables and e_i is the residual error that can be neglected in practice if a large enough number of bases are used. This is the consequence of the Karhunen–Loeve expansion (Yaglom, 1987; Adler, 1990; Kwapien and Woyczynski, 1992; Dougherty, 1999).

Karhunen–Loeve expansion states that $\epsilon_i(t)$ can be decomposed as

$$\epsilon_i(t) = \sum_{l=0}^{m} Z_l \phi_l(t)$$

for uncorrelated Gaussian random variables Z_l and some orthonormal basis $\phi_l(t)$. The algebraic determination of Z_l and $\phi_l(t)$ are left as an exercise. Since ϕ_l and ψ_l are different bases, ϕ_l can be represented as a linear combination of ψ_l. Rewriting ϕ_l in terms of ψ_l will make the Gaussian random variables Z_l correlated.

At the end, we can write model (4.14) as

$$\zeta_i(t) = \sum_{l=0}^{k} X_l \psi_l(t) + e_i(t), \tag{4.16}$$

where X_l are correlated Gaussian random variables.

4.6.3 Parameter Estimation

In practice, functional time series are observed at discrete time points $t_1, t_2, \cdots, t_n$:

$$\zeta_i(t_j) = \mu_i(t_j) + \epsilon_i(t_j), \quad j = 1, \cdots, n.$$

The underlying mean functions $\mu_i(t)$ are estimated as

$$\widehat{\mu}_i(t) = \sum_{l=0}^{k} c_{li} \psi_l(t),$$

where the Fourier coefficients $(c_{0i}, c_{1i}, \cdots, c_{ki})$ for the ith time series is estimated using LSE. If we have p number of time series, it requires p number of LSE separately for each time series. For really large LSE problems, this is computationally very inefficient. A more efficient way is to estimate all the coefficients using a single LSE by solving the following large normal equation:

$$Y_{n\times p} = \Psi_{n\times k} C_{k\times p},$$

where

$$Y_{n\times p} = \begin{pmatrix} \zeta_1(t_1) & \zeta_2(t_1) & \cdots & \zeta_p(t_1) \\ \zeta_1(t_2) & \zeta_2(t_2) & \cdots & \zeta_p(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_1(t_n) & \zeta_2(t_n) & \cdots & \zeta_p(t_n) \end{pmatrix}, \tag{4.17}$$

$$\Psi_{n\times k} = \begin{pmatrix} \psi_0(t_1) & \psi_1(t_1) & \cdots & \psi_k(t_1) \\ \psi_0(t_2) & \psi_1(t_2) & \cdots & \psi_k(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_0(t_n) & \psi_1(t_n) & \cdots & \psi_k(t_n) \end{pmatrix}, \tag{4.18}$$

$$C_{k\times p} = \begin{pmatrix} c_{01} & c_{02} & \cdots & c_{0p} \\ c_{11} & c_{12} & \cdots & c_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \cdots & c_{kp} \end{pmatrix}. \tag{4.19}$$

Subsequently, the coefficients are simultaneously estimated in the least squares fashion as

$$\widehat{C} = (\Psi^\top \Psi)^{-1}\Psi^\top Y.$$

It is possible to discretize the basis ψ_l in such a way that $\Psi^\top\Psi = I_k$. This will make the numerical implementation of the Fourier series expansion computationally much more efficient for big data. The proposed least squares estimation technique avoids using the often used implicit Fourier transform (FT) (Bulow, 2004; Gu et al., 2004; Batchelor et al., 2006). The advantage of the cosine representation is that, instead of recording all the values of time series data, we only need to record $p \cdot (k + 1)$ number of parameters. This is a substantial data reduction, and we may be able to compress fMRI into 5% of the original data while suppressing high-frequency noise (Figure 4.2).

Cosine series representation of fiber tracts. Cosine series representation can also be used in modeling white matter fiber tracts (Chung et al., 2010a). Unlike the nonparametric way of representing white matter fiber connectivity probabilistically, parametric methods can be used to model white matter fibers explicitly. Splines have also been often used for modeling and matching 3D curves (Kishon et al., 1990). Unfortunately, splines are not easy to model and to manipulate explicitly compared to Fourier descriptors, due to the introduction of internal knots. In Clayden et al. (2007), the cubic-B spline is used to parameterize the median of a set of tracts for tract dispersion modeling. Matching two splines with different numbers of knots is not computationally trivial and has been solved using a sequence of ad hoc approaches. In Gruen and Akca (2005), the optimal displacement of two cubic spline curves is obtained by minimizing the sum of squared Euclidean distances. The minimization is nonlinear, so an iterative updating scheme is used. On the other hand, there is no need for any numerical optimization in curve matching in Fourier descriptors due to the nature of the Hilbert space framework. Instead of using the squared distance of coordinates, others have used the curvature and torsion as features to be

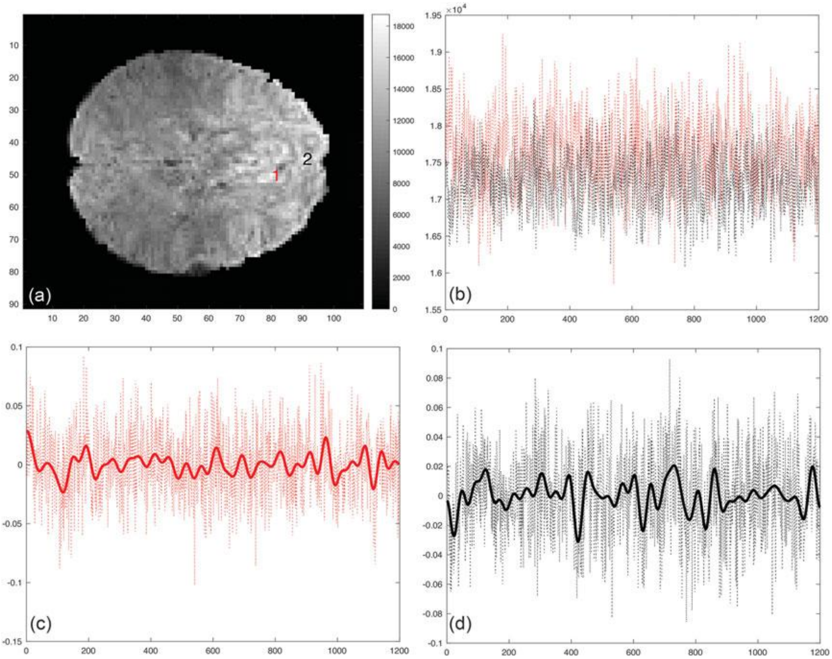


Figure 4.2 (a) Resting-state fMRI at two different voxels at the first time point. (b) Resting-state fMRI at voxel 1 (red) and 2 (black) shown for all 1,200 time points. (c) Normalized and scaled time series at voxel 1 and its cosine series representation with degree $k = 59$. (d) Normalized and scaled time series at voxel 2 and its cosine series representation with degree $k = 59$. It is unclear what optimal degree we should use.

minimized to match curves (Kishon et al., 1990; Gueziec et al., 1997; Corouge et al., 2004; Leemans et al., 2006). Instead of applying the cosine series representation to functional time series, we apply to x-, y-, and z-coordinates separately (Figure 4.3).

4.6.4 Stepwise Model Selection

One major problem in the cosine series representation is that the expansion has to be truncated at some degree. In Fourier descriptor and spherical harmonic representation literature, the issue of the optimal degree has not been addressed properly, and the degree is simply selected based on a prespecified error bound (Gerig et al., 2001; Bulow, 2004; Gu et al., 2004; Shen et al., 2004; Shen and Chung, 2006). This model selection framework for Fourier descriptors was first

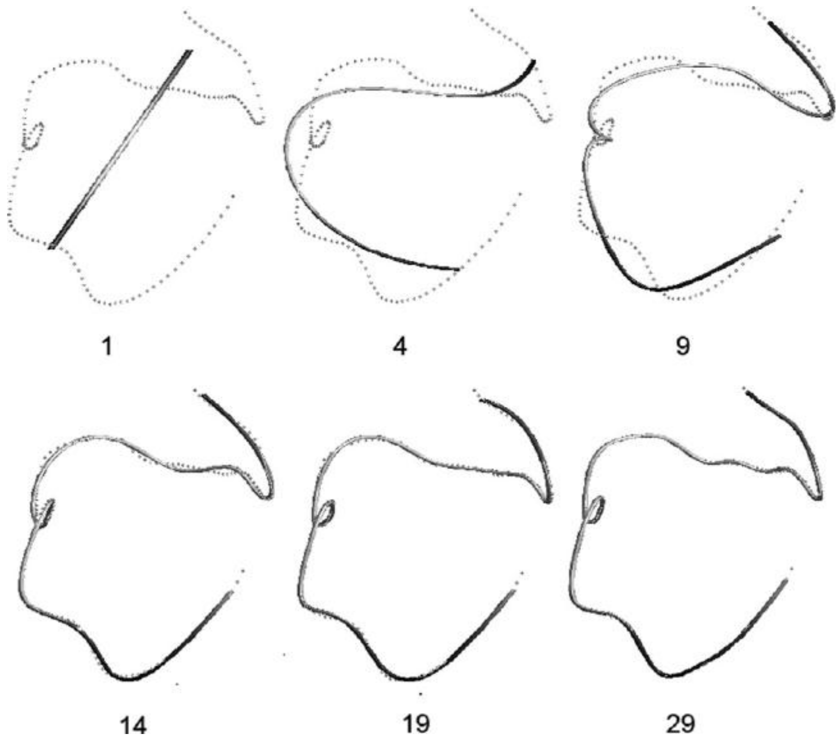


Figure 4.3 Cosine representation of a white matter fiber tract at various degrees. Dots are control points obtained from a streamline-based tractography. The degree 1 representation is a straight line that fits all the control points in a least squares fashion. The degree 19 representation is used through the chapter. It is unclear what optimal degree we should use.

presented in (Chung et al., 2007, 2008a). Although increasing the degree of the representation increases the goodness-of-fit, it also increases the number of estimated coefficients linearly. It is necessary to stop the series expansion at the degree where the goodness-of-fit and the number of coefficients balance out.

Suppose we have the kth degree expansion of signal $\zeta_i(t)$:

$$\zeta_i(t) = \sum_{l=0}^{k} \widehat{c_{li}} \psi_l(t),$$

where $\widehat{c_{li}}$ are the least squares estimation. Assuming up to the $(k-1)$-degree representation is reasonably fitting data well, we determine if adding the k-degree term is statistically significant by testing

$$H_0 : c_{ki} = 0 \text{ vs. } H_1 : c_{ki} \neq 0.$$

Let the kth degree *sum of squared errors* (SSE) be

$$\text{SSE}_k = \sum_{j=1}^{n} \Big[\zeta_i(t_j) - \sum_{l=0}^{k} \widehat{c_{li}} \psi_l(t_j) \Big]^2 .$$

As the degree k increases, SSE decreases until it flattens out. It is reasonable to stop the series expansion when the decrease in SSE is no longer statistically significant. We use the test statistic F given by

$$F = \frac{\text{SSE}_{k-1} - \text{SSE}_k}{\text{SSE}_{k-1}/(n-k-2)} \sim F_{1,n-k-2},$$

which is distributed as the F-distribution with 1 and $n - k - 2$ degrees of freedom under H_0. We compute the F statistic at each degree and stop increasing the degree of expansion if the corresponding p-value first becomes bigger than the prespecified significance, which we can put at $\alpha = 0.01$, for instance. The forward model selection framework hierarchically builds the cosine series representation from lower to higher degree.

4.6.5 Distance between Signals

It is often necessary to measure distance or similarity between functions and time series. Using the cosine series representation, we can determine the optimal distance between the collections of functional time series, which avoids brute-force style numerical optimization schemes often used in the functional data analysis field (Kishon et al., 1990; Gueziec et al., 1997; Ramsay and Silverman, 1997; Gruen and Akca, 2005; Leemans et al., 2006). This simplicity makes the cosine series representation more well suited than more often used splines or other signal filtering techniques (Gruen and Akca, 2005).

With the abuse of notations, we will interchangeably use functional signals to be estimated and their estimation with the same notations when the meaning is clear. Let the cosine series representation of two collections of time series $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ be

$$\boldsymbol{\eta}(t) = \sum_{l=0}^{k} \boldsymbol{\eta}_l \psi_l(t),$$

$$\boldsymbol{\zeta}(t) = \sum_{l=0}^{k} \boldsymbol{\zeta}_l \psi_l(t)$$

where $\boldsymbol{\eta}_l$ and $\boldsymbol{\zeta}$ are the Fourier coefficient vectors. $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ are collections of times series in a vectorial form.

Consider a displacement vector $\mathbf{u} = (u_1, u_2, \cdots, u_p)^\top$ that is required to register $\boldsymbol{\zeta}$ to $\boldsymbol{\eta}$ as close as possible. We will determine an optimal displacement

$\mathbf{u}$ such that the distance between the deformed curve $\boldsymbol{\zeta} + \mathbf{u}$ and $\boldsymbol{\eta}$ is minimized with respect to a certain distance measure ρ. The distance ρ between $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ is defined as the integral of the sum of squared distance:

$$\rho(\boldsymbol{\zeta}, \boldsymbol{\eta}) = \int_0^1 \|\boldsymbol{\zeta}(t) - \boldsymbol{\eta}(t)\|^2 \, dt.$$

The distance ρ can be simplified as

$$\begin{aligned} \rho(\boldsymbol{\zeta}, \boldsymbol{\eta}) &= \int_0^1 \sum_{j=1}^{p} \left[\sum_{l=0}^{k} (\zeta_{lj} - \eta_{lj}) \psi_l(t) \right]^2 dt \\ &= \sum_{j=1}^{p} \sum_{l=0}^{k} (\zeta_{lj} - \eta_{lj})^2. \end{aligned}$$

We have used the orthogonality condition

$$\int_0^1 \psi_l(t) \psi_m(t) \, dt = \delta_{lm}$$

to simplify the expression. It is left as an exercise to show $\rho(\boldsymbol{\zeta}, \boldsymbol{\eta})$ is a proper metric.

Theorem 4.5 *Let* $\mathcal{H}_k = \{\sum_{l=0}^k c_l \psi_l(t) : c_l \in \mathbb{R}\}$ *be the subspace spanned by up to kth basis. Then we have*

$$\arg \min_{u_1, \cdots, u_p \in \mathcal{H}_k} \rho(\boldsymbol{\zeta} + \mathbf{u}, \boldsymbol{\eta}) = \sum_{l=0}^{k} (\boldsymbol{\eta}_l - \boldsymbol{\zeta}_l) \psi_l(t).$$

Proof. Let $\mathbf{u}^*(t)$ be the optimal displacement, which has a form

$$\mathbf{u}^*(t) = \sum_{l=0}^{k} \mathbf{u}_l \psi_l(t)$$

for some unknown parameter vector $\mathbf{u}_l = (u_{l1}, u_{l2}, \cdots, u_{lp})^\top$. Then

$$\rho(\boldsymbol{\zeta} + \mathbf{u}^*, \boldsymbol{\eta}) = \sum_{j=1}^{p} \sum_{l=0}^{k} (\zeta_{lj} + u_{lj} - \eta_{lj})^2,$$

which is an unconstrained positive definite quadratic program with respect to variables u_{lj}. The global minimum always exists and is obtained when $\rho(\boldsymbol{\zeta} + \mathbf{u}^*, \boldsymbol{\eta}) = 0$. Thus, we have $u_{lj} = \eta_{lj} - \zeta_{lj}$. □

The simplicity of Theorem 4.5 is that function registration is done by simply matching the corresponding Fourier coefficients without any sort of numerical optimization as in spline curve matching. In (4.20), the distance between two collections of time series is given as a function of degree k. Thus, the *multiscale distance* that captures both low- and high-frequency similarity can be given by

$$\sum_{k=0}^{K}\sum_{j=1}^{p}\sum_{l=0}^{k}(\zeta_{lj} - \eta_{lj})^2,$$

where K is the preselected degree.

White matter fiber tract registration and clustering. The method can be applied to linearly registering the fiber tracts and used for clustering (see Figure 4.4). The cosine series representation can be used to analyze a

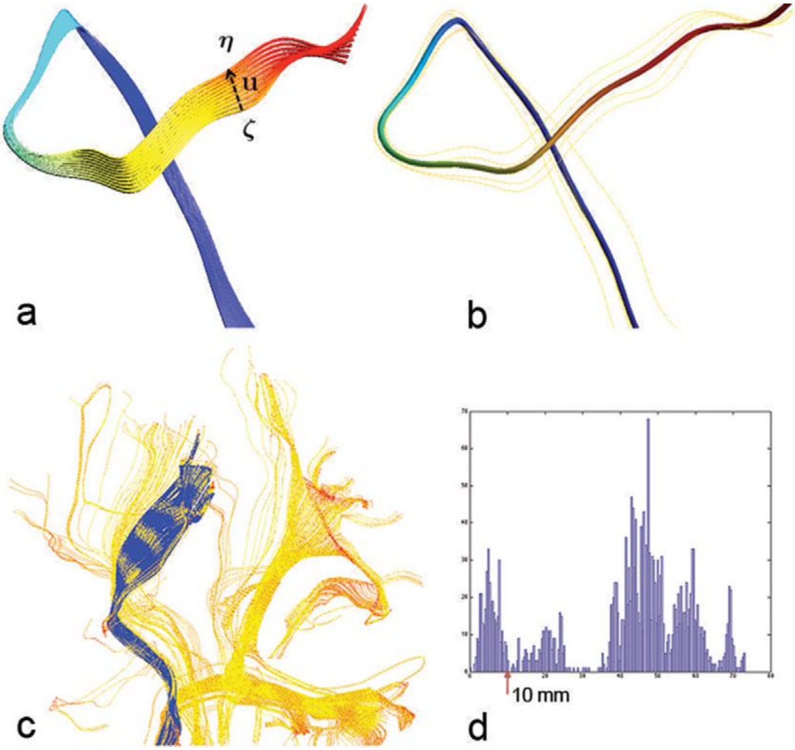


Figure 4.4 (a) The curve $\boldsymbol{\zeta}$ is registered to $\boldsymbol{\eta}$ by the displacement vector field $\mathbf{u}$, which is estimated in the least squares fashion using the cosine series representation. The other intermediate curves are generated by plotting $\boldsymbol{\zeta} + \alpha\mathbf{u}$ with $\alpha \in [0, 1]$ to show how the different amount of displacement deforms the curve $\boldsymbol{\zeta}$. (b) The average of a fiber bundle consisting of five tracts obtained by averaging the cosine coefficients. (c) Fiber tract clustering based on the distance between tracts. (d) The histogram of distances from a single tract (one of blue tracts) to all other tracts. By thresholding the histogram at 10 mm, we cluster tracts.

collection of fiber bundles consisting of similarly shaped curves. The ability to register one tract to another tract is necessary to establish anatomical correspondence for a subsequent population study. Since curves are represented as combinations of cosine functions, the registration will be formulated as a minimization problem in the subspace $\mathcal{H}_k$, which avoids brute-force-style numerical optimization schemes (Kishon et al., 1990; Gueziec et al., 1997; Ramsay and Silverman, 1997; Gruen and Akca, 2005; Leemans et al., 2006). This simplicity makes the cosine series representation better suited than the usual spline representation of curves in subsequent statistical analysis (Gruen and Akca, 2005).

4.6.6 Inference on a Collection of Functional Signals

Based on the idea of computing distance between functional signals by matching coefficients, we can construct the average functional signals of p functional signals $\boldsymbol{\zeta}^1, \cdots, \boldsymbol{\zeta}^p$ by finding the optimal function that minimizes the sum of all discrepancies in subspace $\mathcal{H}_k$:

$$\overline{\boldsymbol{\zeta}}(t) = \arg \min_{\zeta_1, \cdots, \zeta_p \in \mathcal{H}_k} \sum_{j=1}^{p} \rho(\boldsymbol{\zeta}^j, \boldsymbol{\zeta}).$$

The algebraic manipulation can show that the optimum signal is obtained by the average of representation:

$$\overline{\boldsymbol{\zeta}}(t) = \frac{1}{p} \sum_{j=1}^{p} \sum_{l=0}^{k} \boldsymbol{\zeta}_l^j \psi_l(t) = \sum_{l=0}^{k} \overline{\boldsymbol{\zeta}}_l \psi_l(t), \tag{4.20}$$

where $\overline{\boldsymbol{\zeta}}_l$ is the average coefficient vector

$$\overline{\boldsymbol{\zeta}}_l = \frac{1}{p} \sum_{j=1}^{p} \boldsymbol{\zeta}_l^j.$$

This simplicity is the consequence of the Fourier series having the best representation in the Hilbert space. Similarly, we can define the sample variance of p signals and it will turn out to be the cosine representation with the coefficient vector consisting of the sample variance of p coefficients. The construction of the sample variance of p signals should be fairly straightforward, and we will not go into detail.

Given another collection of functional signals $\boldsymbol{\eta}^1, \cdots, \boldsymbol{\eta}^q$, we can perform statistical inference on the equality of functional signals in the two populations.

The null hypothesis of interest is

$$H_0 : \overline{\boldsymbol{\zeta}} = \overline{\boldsymbol{\eta}}. \tag{4.21}$$

Here we again abused the notation so we are testing the equality of mean representations of populations. From the very property of the Fourier series in Hilbert space, the uniqueness of the cosine series representation is guaranteed so the two representations are equal if and only if the coefficients vectors match. Therefore, the equivalent hypothesis to (4.21) is given by

$$H_0' : \overline{\boldsymbol{\zeta}}_1 = \overline{\boldsymbol{\eta}}_1, \cdots, \overline{\boldsymbol{\zeta}}_k = \overline{\boldsymbol{\eta}}_k.$$

Obviously this is a multiple comparisons problem. Under the Gaussian assumption in (4.16), testing the equality of the mean coefficient vector can be done using Hotelling's T-square statistic. For correcting for the multiple comparisons, the Bonferroni correction can be used.

4.6.7 Gibbs Phenomenon

Limitation of cosine series representation. A downside of using Fourier descriptors is that they are not local and it is not possible to make a statement about a specific portion of the functional signal. Although the Fourier coefficients are global and mainly used for globally classifying shapes (Shen et al., 2004), it is still possible to obtain local shape information and make a statement about local shape characteristics (Chung et al., 2007).

Splines have also been widely used for modeling functional signals (Kishon et al., 1990; Gruen and Akca, 2005; Clayden et al., 2007). Unfortunately, splines are not easy to model and to manipulate explicitly compared to Fourier descriptors, due to the introduction of internal knots. In Clayden et al. (Corouge et al., 2004; Clayden et al., 2007), the cubic-B spline is used to parameterize the median of a set of functions. Matching two splines with different numbers of knots is not computationally trivial and has been solved using a sequence of ad hoc approaches. In Gruen et al. (Gruen and Akca, 2005), the optimal displacement of two cubic spline curves is obtained by minimizing the sum of squared Euclidean distances. The minimization is nonlinear, so an iterative updating scheme is used. On the other hand, there is no need for any numerical optimization in obtaining the matching in our method due to the very nature of the Hilbert space framework. Instead of using the squared distance of coordinates, others have used the curvature and torsion as features to be minimized to match curves (Kishon et al., 1990; Gueziec et al., 1997; Leemans et al., 2006).

The *Gibbs phenomenon* (ringing artifacts) often arises in Fourier series expansion of discontinuous data. It is named after American physicist Josiah Willard Gibbs. In representing piecewise continuously differentiable data using the Fourier series, the overshoot of the series happens at a jump discontinuity. The overshoot does not decease as the number of terms increases in the series expansion, and it converges to a finite limit called the Gibbs constant. The Gibbs phenomenon was first observed by Henry Willbraham in 1848 (Wilbraham, 1848) but it did not attract any attention at that time. Then a Nobel Prize laureate, Albert Michelson, constructed an harmonic analyzer, one of the first mechanical analogue computers, which was used to plot Fourier series, and observed the phenomenon. He thought the phenomenon was caused by mechanical error, but Josiah Willard Gibbs correctly explained the phenomenon as mathematical in 1899. Gibbs rediscovered the phenomenon in 1898 (Gibbs, 1898). Later, mathematician Maxime Bocher named it the Gibbs phenomenon and gave a precise mathematical analysis in 1906 (Bocher, 1906). The Gibbs phenomenon associated with spherical harmonics were first observed by Herman Weyl in 1968. The history and the overview of Gibbs phenomenon can be found in the literature (Foster and Richards, 1991; Jerri, 1998).

There are few available techniques for reducing Gibbs phenomenon (Gottlieb and Shu, 1997; Brezinski, 2004). Most techniques are a variation on some sort of kernel methods. For instance, consider Fejer kernel K_n defined as

$$K_n(u) = \frac{1}{n}\sum_{j=0}^{n-1} D_j(u),$$

where D_j is the Dirichlet kernel

$$D_j = \sum_{k=-j}^{j} e^{iku}.$$

Then it can be shown that

$$K_n(u) = \frac{1}{n}\left(\frac{\sin\frac{nu}{2}}{\sin\frac{u}{2}}\right)^2.$$

The kernel is symmetric and positive. Then it can be shown that

$$K_n * f \to f$$

for any, even discontinuous, $f \in \mathcal{L}^2[-\pi,\pi]$ as $n \to \infty$. It has the effect of smoothing the discontinuous signal f and in turn the convolution will not exhibit the ringing artifacts for sufficiently large n. Particularly related to Fourier and spherical harmonic descriptors is an exponential weighting scheme that we have introduce (Chung et al., 2007, 2008a). By weighting Fourier coefficients with exponentially decaying weights, the series expansion can converge faster and reduce the Gibbs phenomenon significantly.

Instead of the kth degree expansion (4.15), we define the weighted Fourier expansion as

$$\sum_{l=0}^{k} e^{-\lambda_l \sigma} \langle f, \psi_l \rangle \psi_l \tag{4.22}$$

for some smoothing parameter σ. Then it can be shown that (4.22) is the finite series expansion of heat kernel smoothing $K_\sigma * f$, where the heat kernel is defined as

$$K_\sigma(t,s) = \sum_{l=0}^{\infty} e^{-\lambda_l \sigma} \psi_l(t) \psi_l(s).$$

The expansion (4.22) can be further shown to be the finite approximation to the solution of heat diffusion

$$\frac{\partial}{\partial \sigma} g = \Delta g, \; g(t, \sigma = 0) = f(t).$$

Since the weighting scheme makes the expansion converge to heat diffusion, the estimation at the jump discontinuity is smoothed out reducing the Gibbs phenomenon.

4.7 Correlating Functional Signals

We can also use correlations to measure distance between functions. The concept of correlation here can be viewed as the generalization of Pearson correlation that is applied to discrete vector data to functional data. Consider functional signal

$$\zeta_1(t), \cdots, \zeta_p(t) \in L^2[0,1].$$

Definition 4.7 *The mean of functional signal ζ_i over time is given by*

$$\mathbb{E}_t \zeta_i = \int_0^1 \zeta_i(t)\, dt.$$

The cross-covariance *between functional signals ζ_i and ζ_j over interval* $[0,1]$ *is then given by*

$$\mathbb{V}_t(\zeta_i, \zeta_j) = \mathbb{E}_t\big[(\zeta_i - \mathbb{E}_t \zeta_i)(\zeta_j - \mathbb{E}_t \zeta_j)\big].$$

The variance *of* ζ_i *is*

$$\mathbb{V}_t \zeta_i = \mathbb{V}_t(\zeta_i, \zeta_i) = \int_0^1 \left[\zeta_i(t) - \mathbb{E}_t \zeta_i(t)\right]^2 dt.$$

The cross-correlation coefficient between functional signals ζ_i *and* ζ_j *over interval* $[0,1]$ *is*

$$\rho_{ij} = \frac{\mathbb{V}_t(\zeta_i, \zeta_j)}{\sqrt{\mathbb{V}_t(\zeta_i)\mathbb{V}_t(\zeta_j)}}.$$

Often we subtract $\mathbb{E}_t \zeta_i$ from functional signal ζ_i, i.e. $\zeta_i - \mathbb{E}_t$ such that ζ_i is centered. This operation is often done to normalize signals since subjects have different baseline average signals. From now on, we will simply assume $\mathbb{E}_t \zeta_i = 0$. If not, simply subtract by its mean. To reduce the confusion, let $\eta_i = \zeta_i - \mathbb{E}_t$ be the centered data of ζ_i. Consider the cosine series representation of centered η_i.

$$\eta_i = \sum_{l=0}^{k} c_{li} \psi_l(t) + e_i(t),\ t \in [0,1].$$

$e_i(t)$ is the residual function of the fit. If we use sufficiently large k, it is expected that e_i is small and ignorable.

The covariance of η_i and η_j is given by

$$\begin{aligned}\mathbb{V}_t(\eta_i, \eta_j) &= \int_0^1 \eta_i(t)\eta_j(t)\, dt \\ &\approx \sum_{l,m=0}^{k} c_{li} c_{mj} \int_0^1 \psi_l(t)\psi_m(t)\, dt \\ &= \sum_{l=0}^{k} c_{li} c_{lj} = \mathbf{c}_i^\top \mathbf{c}_j,\end{aligned}$$

where $\mathbf{c}_i = (c_{0i}, c_{1i}, \cdots, c_{ki})^\top$. The variance of η_i is then given by

$$\mathbb{V}_t \eta_i = \int_0^1 \eta_i^2(t)\, dt \approx \sum_{l=0}^{k} c_{li}^2 = \mathbf{c}_i^\top \mathbf{c}_i.$$

We used the fact that ψ_l are orthonormal, i.e.,

$$\int_0^1 \psi_l(t)\psi_m(t)\, dt = \delta_{lm}.$$

The cross-correlation between functional signals η_i and η_j is then given by

$$\rho_{ij} = \frac{\mathbf{c}_i^\top \mathbf{c}_j}{[\mathbf{c}_i^\top \mathbf{c}_i \mathbf{c}_j^\top \mathbf{c}_j]^{1/2}}.$$

If we let $\mathbf{b}_i = \mathbf{c}_i / \sqrt{\mathbf{c}_i^\top \mathbf{c}_i}$, the correlation is simplified as

$$\rho_{ij} = \mathbf{b}_i^\top \mathbf{b}_j.$$

Let $B_{k \times p} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_p]$ be the matrix of scaled Fourier coefficients. Then the correlation matrix $\rho = (\rho_{ij})$ is given by $\rho = B^\top B$. This will provide a faster but approximate computation of large-scale correlation matrices. Note that (4.23) is exact only if k is larger than the number of sampling points n. As k increases, the accuracy is expected to increase (Figure 4.5).

Example 4.2 *Consider fMRI time series η_1 and η_2 obtained in two voxels 1 and 2 in the Figure 4.2 example. The Pearson correlation between η_1 and η_2 is*

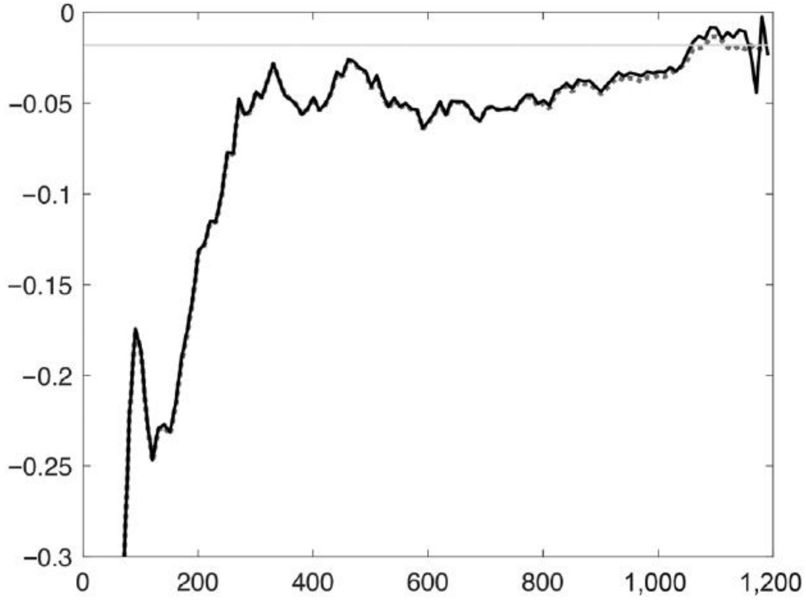


Figure 4.5 The straight line is the correlation between resting-state fMRI in two voxels. The dotted line is the correlation between the cosine series representations. The solid line is using the product of Fourier coefficients $\mathbf{b}_i^\top \mathbf{b}_j$. As the degree of expansion increases, they converge to each other.

−0.018. The baselines are $\mathbb{E}_t\eta_1 = 1{,}763.0$ *and* $\mathbb{E}_t\eta_2 = 1{,}724.8$. *Now perform degree 59 cosine series expansion on the mean subtracted signals:*

$$\eta_1 - 1763.0 = \sum_{l=0}^{59} c_{l1}\psi_l(t) + e_1(t)$$

$$\eta_2 - 1724.8 = \sum_{l=0}^{59} c_{l2}\psi_l(t) + e_2(t), \tag{4.23}$$

where $e_1(t)$ *and* $e_2(t)$ *are the residual model fit. In this example,*

$$\mathbb{E}_t e_1 = 3.05 \cdot 10^{-7}, \quad \mathbb{E}_t e_2 = 5.72 \cdot 10^{-6}.$$

The Pearson correlation becomes −0.3329.

4.8 Thresholding Correlation Networks

The majority of brain network analyses have been based on thresholding correlation edge weights in detecting focal regions of correlated voxels (Cao and Worsley, 1999b; Koch et al., 2002). On the other hand, Worsley et al. (2005a) used the singular value decomposition (SVD) in showing that SVD is better at detecting extensive regions of correlated voxels compared to the traditional method of simple correlation thresholding. Let $X_{n\times p} = (x_{ij})$ be the matrix of p regions and n subjects. Unless it is large sample size studies, we expect the *large p small n problem.* We assume X to be centered by subtracting their mean value. We further assume that each column of X is normalized by dividing by its root sum of squares, so that the diagonal elements of the cross-correlation matrix $\Sigma_{p\times p} = X'X$ is 1. The SVD of Σ is as follows:

$$\Sigma = UWU',$$

where U is an orthonormal matrix and W is a diagonal matrix of component weights.Worsley et al. (2005a) proposed to estimate Σ by setting the smaller weights in W to be zero. This is exactly the principal component analysis or partial least squares (PLS) (McIntosh et al., 1996; McIntosh and Lobaugh, 2004). PLS is similar to PCA, but the solutions of PLS are constrained to be part of the covariance structure. Since p can possibly reach upward of few million voxels, the computational burden of finding SVD of Σ can be prohibitive in small computers. Worsley et al. (2005a) proposed to bypass the problem by matrix decompositions. Afterward, the statistical inference is done either using permutation tests (Nichols and Holmes, 2002) or the random field theory (Worsley et al., 1998; Cao and Worsley, 1999b). Since the brain networks are known to be sparse and highly clustered (Achard and Bullmore, 2007; He et al., 2007), it is reasonable to incorporate the sparsity of network structures into PCA further. There have been various attempts in incorporating spasticity in PCA using LASSO in statistics (Jolliffe et al., 2003; Zou et al., 2006). LASSO is a widely used variable selection technique that produces sparse models (Tibshirani, 1996). It is based on the observation that PCA can be reformulated as the optimal solution of a regression so that LASSO can be integrated into the regression.

The main limitation of connectivity analyses based on correlation or covariance matrices is that it fails to explicitly factor out the confounding effect of other regions. To remedy this limitation, partial correlation has been naturally introduced in factoring out the dependencies of other regions (Marrelec et al., 2006; He et al., 2007) or eliminating the effect of the experimental design (McIntosh et al., 1996). Since the partial correlation corresponds to the off-diagonal entries of the inverse covariance matrix, sparse PCA can be used to the inverse covariance matrix. A similar frameworks found applications in image classification (Berge et al., 2007), gene expression (Dobra et al., 2004), flow cytometry data (Friedman et al., 2008), and functional brain network modeling (Huang et al., 2009, 2010).

5
Big Brain Network Data

In this chapter, we explore the main characteristics of big brain network data that offer unique computational challenges. The brain networks are biologically expected to be both sparse and hierarchical. Such unique characterizations put specific topological constraints onto statistical approaches and models we can use effectively. We explore the limitations of the current approaches used in the field, offer alternative approaches, and explain new challenges as well as provide the computional solutions.

5.1 Big Data

Wikipedia defines *big data* as data sets that are so large or complex that traditional data processing application software is inadequate to deal with them (`en.wikipedia.org/wiki/Big_data`). Big data are not just about the size of the data, although that is the main obstacle of using traditional statistical approaches. Big data usually include data sets with sizes beyond the ability of standard software tools to process and analyze within a reasonable time limit. Even 100 MB of data can be big if existing computing resources can only handle 1 MB of data at a time. Thus, the size of the data is a *relative* quantity with respect to the available computing resources.

If we pick any article in big data literature these days, chances are that we will often encounter hardware solutions to solving big data problems. They often suggest increasing more central processing units (CPU) or graphical processing units (GPU) and emphasize the need for cluster or parallel computing. For instance, Boubela et al. (2016) suggest to use parallel computing as a way to compute large-scale Pearson correlation coefficients for 390 GB of data in the HCP but did not suggest any other simpler algorithmic approaches that can be implemented in a limited computing resource environment. Simply adding more hardware is costly and not necessarily an effective strategy for big data. Such hardware approaches often do not provide a venue for more interesting statistical problems. Further, the access to fast computational resources is not necessarily given to everyone. Many biological laboratories still do not have technical expertise of using cluster or parallel computing. Therefore, it is often necessary to develop more algorithmic and statistical approaches in addressing big data at least for biological sciences.

Big brain image and network data offer unique computational challenges. The brain networks are biologically expected to be both sparse and hierarchical. Such unique characterizations put specific topological constraints onto statistical approaches and models we can use effectively (Chung, 2018).

5.1.1 Large-Scale Brain Images

Many big data sets introduce unique computational and statistical challenges that include scalability, storage bottleneck, data representation visualization, and computation mostly related to sample sizes (Fan et al., 2014). However, the challenges in big brain imaging data sets such as HCP and the Alzheimer's Disease Neuroimaging Initiative (ADNI; `adni.loni.usc.edu`) are slightly different.

The majority of functional and structural connectivity studies in brain imaging are usually performed following the standard analysis framework (Hagmann et al., 2007; Gong et al., 2009; Fornito et al., 2010; Zalesky et al., 2010). From 3D whole brain images, n regions of interest (ROI) are identified and serve as the nodes of the brain network. Measurements at ROIs are then correlated in a pairwise fashion to produce the connectivity matrix of size $p \times p$. The connectivity matrix is then thresholded to produce the adjacency matrix consisting of zeros and ones that define the link between two nodes. The binarized adjacency matrix is then used to construct the brain network. However, for a large number of nodes, this brute force approach has a serious computational bottleneck of manipulating a huge number of connections and storing them. Even if we solve the computational problem, biomedical interpretation of network will be difficult with the huge number of links. For example, for 3×10^5 voxels in an image, we can possibly have a total of 9×10^{10} edges in the graph.

Further, there are substantially more number of voxels (p) per image than the number of images (n) in the data sets. Even at 3 mm low resolution, fMRIs have more than 25,000 voxels (Chung et al., 2017a). Unless the data set consists of more than 25,000 images, brain imaging is often the problem of *small-n large-p*, which is different from the usual big data setting where n is often big. HCP and ADNI have n in the range of thousands, far smaller than the number of voxels.

Traditionally, numerical accuracy has been less of concern in brain imaging particularly due to spatial and temporal smoothing often done in images to smooth out various image processing artifacts and physiological noises. Due to the increased sample size and the central limit theorem, which is further reinforced by smoothing, the statistical distribution of the data might become less of a concern in big imaging data (Salmond et al., 2002).

In the traditional mass univariate approaches (Worsley et al., 1992), where statistical inference is done at each voxel, the problem of small-n large-p is not critical. Further, spatial smoothing has the effect of reducing the number of *resolution element* (RESEL), so we have far fewer effective p (Worsley et al., 1992). Smoothing also reduces the effect of image registration errors and high-frequency noise. Gaussian kernel smoothing introduces continuous hierarchical structure through scale space (Worsley et al., 1996a). However, small-n large-p problems become critical in brain network modeling, where we need to correlate different voxels. In the small-n large-p setting, the sample covariance and correlation matrices are no longer positive definite. Subsequently, up to $p - n$ nodes are statistically dependent although there might be *no* true dependency at all. Thus, there is need to constrain the covariance or correlation matrices by regularization methods such as sparse network models. Unfortunately, for large p, many sparse models have severe computational bottlenecks (Chung et al., 2015a).

There begin to emerge large-scale brain networks with more than 25,000 nodes, where each voxel is taken as a network node (Figure 5.1) (Eguíluz et al., 2005; Hagmann et al., 2007; Chung et al., 2017a; Taylor et al., 2017). The size of such large-scale brain networks can easily match publicly available network data such as the Stanford Large Network Dataset (`snap.stanford.edu/data`). In such large-scale networks, the small-n large-p problem will be more severe. This type of big data requires more *scalable* and *robust* solutions, possibly using sparse penalties.

Many traditional statistical methods that perform well for reasonable sample size do not scale well with big data. Many likelihood or L_1-norm optimization techniques do not scale well with big data. Methods should likely to scale linearly or at least polynomially to even able to compute in a tolerable time limit. Any method that scales exponentially with increased sample size is not going to be useful. Permutation test is one such method. Any method that requires the complete data set as an input is not going to be useful as well. The following are the desirable properties of efficient computational methods for big data.

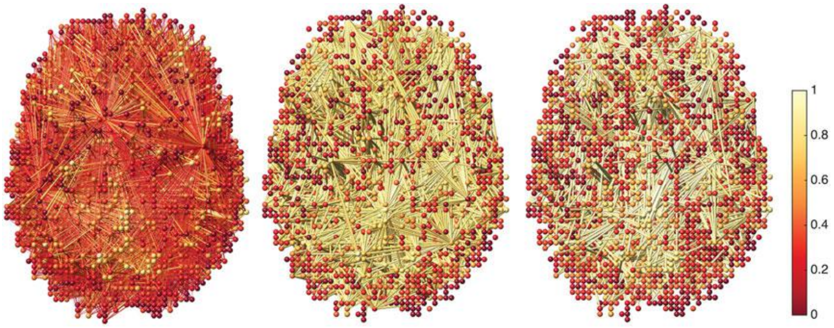


Figure 5.1 Left: Dense fMRI correlation network consisting of more than 25,000 nodes (Chung et al., 2017a). The network is so dense, simply displaying all the nodes and edges of the network is not very informative. It is necessary to represent such a dense network more sparsely. The sparse correlation network model with sparse parameters $\lambda = 0.7$ (middle) and $\lambda = 0.8$ (right). It can be shown that they form a nested hierarchy called the graph filtration.

1. An effective method should be unbiased and sufficiently fast. The *iterative residual fitting* algorithm, originally developed for estimating more than 20,000 spherical harmonic coefficients per brain, may offer one such solution (Chung et al., 2008a). The algorithm sequentially breaks down one gigantic problem into smaller problems in an iterative fashion.
2. An effective method should require only a small subset of the full data in accurately estimating the underlying model. Online algorithms, which we will discuss later, are such methods.

5.1.2 Large-Scale Brain Networks

Purely data-driven approaches for large-scale brain networks are not going to be computationally efficient or effective. It is often necessary to incorporate the first-order principles of brain networks into models to possibly reduce computational bottlenecks. Large-scale brain networks are often characterized by sparsity and hierarchy (Chung, 2018). The sparsity and hierarchy are highly relevant topological characterizations to other types of big network data such as social networks (Christakis and Fowler, 2007), World Wide Web (WWW) (Adamic, 1999), and genomic regulatory networks (Luscombe et al., 2004). Given any type of real-world network, it is unlikely that all the nodes are densely connected to each other. It is expected that the network will have sufficient sparsity. Many large-scale networks such as social networks and WWW show a scale-free characteristic, which is the main characteristic of hierarchical networks. Although we don't expect all networks to be hierarchical or sparse, these aspects of brain network should be applicable to other big network data. In this section, we will explore two main characterizations of brain networks that should be utilized in even small data settings. We will further explore various statistical challenges related to such characterizations.

5.2 Sparsity

At the microscopic level, the activation of cortical neurons in the brain show *sparse* and widely distributed patterns (Histed and Reid, 2009). At the macroscopic level, DTI can produce up to a half-million white matter fiber tracts per brain. Even then, not every part of the brain is anatomically connected to other parts of the brain but sparsely connected (Chung et al., 2017b). This can be seen from Figure 5.2, where the brain is parcellated into 116 disjoint regions and the number of white matter fiber tracts passing between the regions is used in constructing the structural connectivity matrix (Chung et al., 2017b). Even though the white matter fibers are very dense, the resulting connectivity matrix is sparse. For 116×116 connectivity matrix, 60% of entries are zeros. As we increase the number of parcellations, the sparsity increases while the total degree of all nodes decreases (Figure 5.3). Note the degree of nodes counts the number of connections at a node. Thus, it also measures the sparsity of the network.

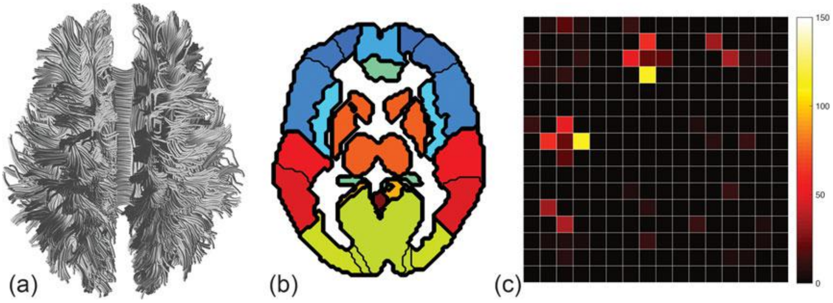


Figure 5.2 (a) White matter fiber tracts obtained from a tractography algorithm. (b) The brain is parcellated into 116 disjoint regions. (c) Connectivity matrix showing how each region is connected to other regions. Even thoroughly fiber tracts are very dense, the resulting connective matrix is always sparse since not every part of brain is connected to each other (Chung et al., 2017b).

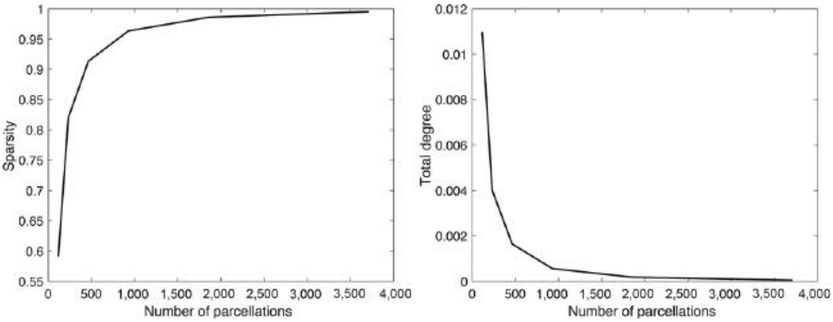


Figure 5.3 Left: plot of sparsity over the number of parcellations. The sparsity is measured as the ratio of zero entries over all entries in the connectivity matrix. Right: plot of total degree of nodes over the number of parcellations. The vertical axis measures the ratio of the total number of connections over every possible connection. The plots all show the sparse nature of brain networks at any spatial scale.

In fMRI studies, functional connectivity, which measures the dependency of brain activity in one region to another region, is often measured by correlation, covariance or spectral coherence of fMRI time series. Since the brain does not activate everywhere simultaneously (Chung et al., 2017a), functional connectivity is also expected to be not dense but sparsely clustered. It is reasonable to assume both functional and anatomical brain networks are sparsely connected at the both microscopic and macroscopic levels. Thus, there is strong biological justification for modeling brain networks sparsely.

The small-n large-p problem in brain imaging often produces underdetermined models with an infinite number of possible solutions. Such problems are usually remedied by regularizing the systems with additional sparse penalties. Sparse models used in brain imaging include compressed sensing (CS) (Lee et al., 2011c), sparse correlations (Chung et al., 2017a), LASSO (Huang et al., 2009; Lee et al., 2011c), sparse canonical correlations (Avants et al., 2010), and graphical-LASSO (Huang et al., 2009; Chung et al., 2015a). Most of these sparse models require optimizing $L1$-norm penalties, which has been the major computational bottleneck for solving large-scale problems in brain imaging. Thus, almost all sparse brain network models have been restricted to a few hundreds nodes or less. Probably the largest number of features used in any sparse model in the brain imaging literature is 2,527 MRI features used in a LASSO model for Alzheimer's disease (Xin et al., 2015) is probably the largest number of features used in any sparse model in the brain imaging literature. Recently, more scalable large-scale sparse brain network models, where each voxel is a network node, are beginning to emerge (Chung et al., 2017a). For such large-scale network construction, faster scalable algorithms are needed.

There are few previous studies at speeding up the computation for sparse models. By identifying block diagonal structures in the estimated inverse covariance matrix, it is possible to reduce the computational burden in the penalized log-likelihood method (Mazumder and Hastie, 2012). In Chung et al. (2017a), the computational bottleneck of $L1$-optimization is overcome by simplifying the sparse network problem into an orthogonal design in LASSO (Tibshirani, 1996). Other promising methods include a constrained L_1-minimization estimator (CLIME) (Wang et al., 2016a) and faster computations for graphical-LASSO (Witten et al., 2011), although they have not yet been applied to large-scale brain networks.

Any sparse brain network model is usually parameterized by a tuning parameter that controls the sparsity of the solution. Increasing the sparse parameter makes the solution more sparse. Thus, sparse models are inherently multiscale, where the scale of the model is determined by the sparsity. Many existing sparse network models use a fixed parameter λ that may not be optimal in other data sets or studies. Depending on the choice of the sparse parameter, the final network structure will be different (Lee et al., 2012; Chung et al., 2015a). There is a need to develop a multiscale sparse network model that provides consistent analysis results and interpretation regardless of the choice of parameter (Chung et al., 2015a, 2017a).

5.3 Hierarchy

Brain networks are fundamentally *multiscale*. An intuitive and palatable biological hypothesis is that brain networks are organized into *hierarchies* (Betzel and Bassett, 2017). A brain network at any particular sale might be subdivided into subnetworks, which can be further subdivided into smaller subnetworks in an iterative fashion. There have been various attempts at modeling brain networks at multiple scales (Lee et al., 2012; Chung et al., 2015a, 2017a; Betzel and Bassett, 2017). Unfortunately, many multiscale models give raise to conflicting topological structures of the networks from one scale to the next. For instance, the estimated modular structure in the multiscale community detection problem usually do not have continuity over different resolution parameters (Betzel and Bassett, 2017). The topological structure of parcellation at one particular scale may not carry over to different scales

(Zalesky et al., 2010; Betzel and Bassett, 2017). There is a need to develop a hierarchical parcellation scheme that provide a consistent network analysis results and interpretation regardless of the choice of scale.

5.3.1 Hierarchical Structural Parcellation

It is possible to come up with a new hierarchical parcellation scheme based on the Courant nodal domain theorem (Courant and Hilbert, 1953). The method is related to graph cuts (Shen et al., 2010) and spectral clustering (Craddock et al., 2012; Pepe et al., 2015) based parcellation schemes previous used in parcellating the resting-state functional magnetic resonance images. However, unlike previous parcellation approaches, it is possible to provide hierarchical nestedness and, thus, preserve topology across different spatial resolutions. This can be achieved through Courant nodal domain theorem (see Figure 5.4).

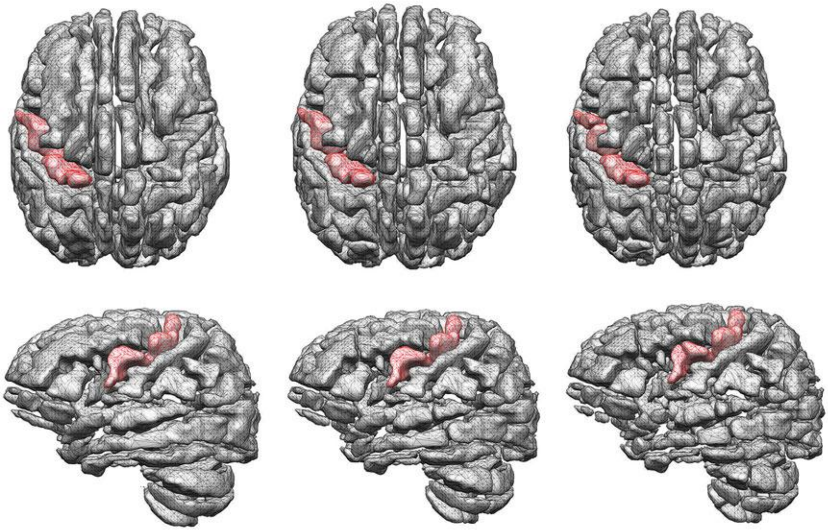


Figure 5.4 Left: anatomical automatic labeling (AAL) parcellation with 116 predefined ROI. Each parcellation is displayed as a disconnected mesh surface. Each ROI serves as a node in a 116 × 116 connectivity matrix. For visualization purposes, we have added artificial empty gaps between mesh surfaces. The red region is the left precentral gyrus. Middle: the second layer of the hierarchical parcellation with 2 × 116 regions. Each AAL parcellation is subdivided into two disjoint regions. Right: the third layer of the hierarchical parcellation with 4 × 116 regions.

Courant nodal domain theorem. For Laplacian Δ in a compact domain $\mathcal{M} \subset \mathbb{R}^3$, consider eigenvalues

$$0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \cdots$$

and eigenfunctions $\psi_0, \psi_1, \psi_2, \cdots$ satisfying

$$\Delta \psi_j(p) = \lambda_j \psi_j(p).$$

We then have $\psi_0(p) = 1/\sqrt{\mu(\mathcal{M})}$, where $\mu(\mathcal{M})$ is the volume of $\mathcal{M}$. From the orthogonality of eigenfunctions, we have

$$\int_{\mathcal{M}} \psi_0(p)\psi_1(p)\, d\mu(p) = 0.$$

Thus, ψ_1 must be take positive and negative values (see Figure 5.5). The Courant nodal domain theorem (Courant and Hilbert, 1953) further states that ψ_1 divides $\mathcal{M}$ into two disjoint regions by the nodal surface boundary

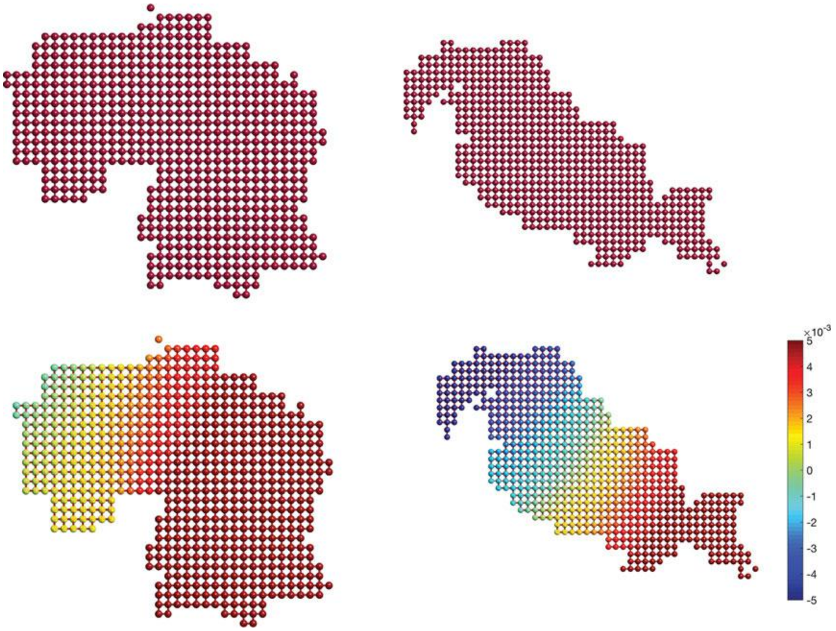


Figure 5.5 Top: a 3D binary image is converted to a 3D graph based on the 18-connected neighbor scheme. Here only a 2D image slice result is shown. Bottom: the Fidler's vector, which is the first nontrivial eigenvector of the graph Laplacian, is then computed. By clustering the graphs depending on the sign of the Fidler's vector, we can split the graph into two disjoint regions.

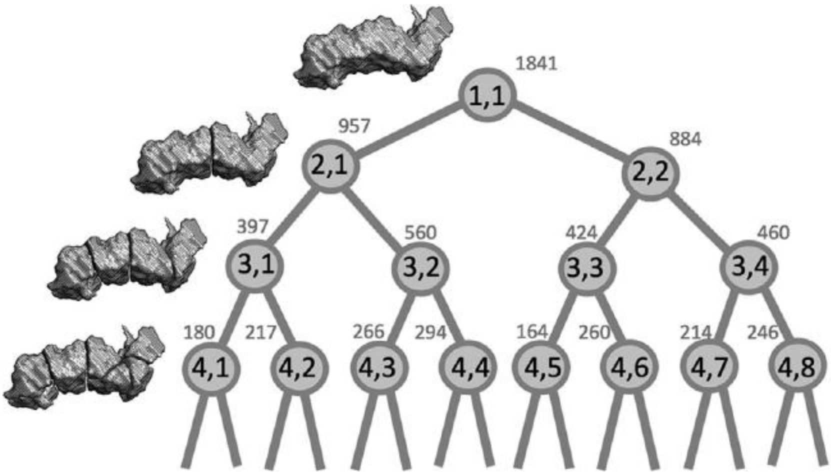


Figure 5.6 The hierarchical parcellation of an AAL ROI (left precentral gyrus shown in Figure 5.7) up to the fourth layer. At the fourth layer, we have 2^3 sub-parcellations. The hierarchical parcellation is represented as a binary tree and indexed using two numbers in computer implementation (layer, indexing for parcellation). The root node (1,1) is the AAL parcellation. At the next layer, nodes (2,1), (2,2) correspond to the partition of the AAL parcellation.

$\psi_1(p) = 0$. When the domain is discretized as a 3D graph, the second eigenfunction ψ_1 is called the Fiedler vector. It is often used in spectral clustering and graph cuts (Shen et al., 2010; Chung et al., 2011b). Applying iteratively the nodal domain theorem, we can hierarchically partition $\mathcal{M}$ in a nested fashion. Once domain $\mathcal{M}$ is divided into $\mathcal{M}_1$ and $\mathcal{M}_2$ based on the sign of the second eigenfunction, we iteratively recompute the second eigenfunction of Laplacian restricted to $\mathcal{M}_1$ and $\mathcal{M}_2$. Then we proceed with subpartitioning $\mathcal{M}_1$ and $\mathcal{M}_2$ further (see Figure 5.6).

The Courant nodal domain theorem can be discretely applied to the AAL parcellation as follows. We first convert the binary volume of each parcellation in AAL into a 3D graph by taking each voxel as a node and connecting neighboring voxels. Using the 18-connected neighbor scheme, we connect two voxels only if they touch each other on their faces or edges. If voxels are only touching at their corner vertices, they are not considered as connected. This results in an adjacency matrix and the 3D graph Laplacian. The computed Fiedler vector is then used to partition each AAL parcellation into two disjoint regions (Figures 5.4 and 5.7). For each disjoint subregion, we further recompute the Fiedler vector restricted to the subregion. This binary partition

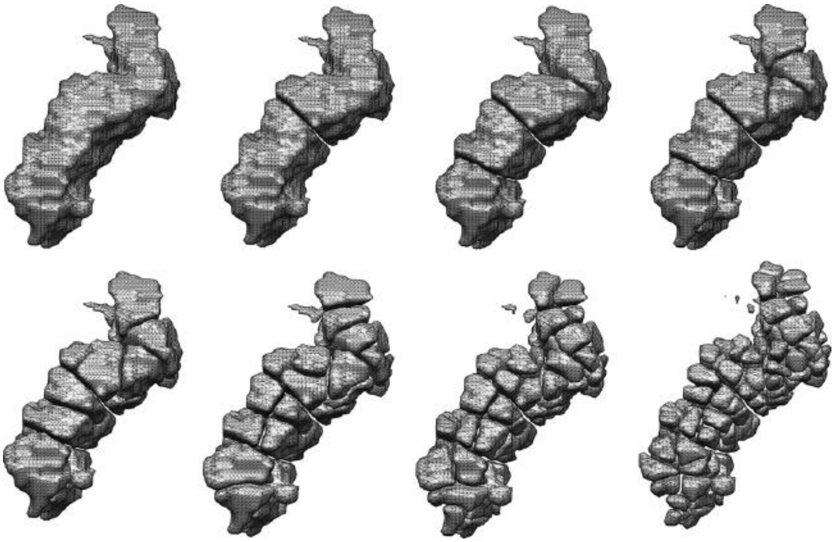


Figure 5.7 Hierarchical parcellation of the left precentral gyrus (shown in Figure 5.4) up to the eighth layer. At the eighth layer, we have $2^{8-1} = 128$ parcellations of the gyrus. The hierarchical parcellation continues until every voxel is a parcellation.

process iteratively continues until all the partitions are voxels. We are doubling the number of parcellations at each iteration. There are a total of $p = 116$ parcellations in layer 1 and $2 \cdot 115$ parcellations in layer 2. At the ith layer, there are $2^{i-1} \cdot 116$ parcellations. This way, we can construct 20-layer nested hierarchical parcellations all the way to the voxel level. Since the number of voxels is not uniform across AAL parcellations, we are *approximately* doubling the number of parcellations at each iteration.

It is possible to hierarchically parcellate the brain regions into small subregions. This requires first converting the binary volume of each parcellation into an adjacency matrix, which represents how each voxel is connected to other voxels. Using the 18-connected neighbor scheme, we define two voxels are connected only if they touch each other on their faces or edges. If they are only touching at their corner vertices, they will not be considered as connected. Figure 5.5 shows the result of binary voxels converted to graphs on two representative slices. Depending on the scale of parcellation, the parameters of graph, which characterize graph topology, vary considerably up to 95% (Fornito et al., 2010; Zalesky et al., 2010). Thus, there is a need for parcellation-free brain network node identification methods.

5.3.2 Hierarchical Structural Connectivity

At the each layer of the hierarchical parcellation (Chung et al., 2018b), we count the total number of white matter fiber tracts connecting parcellations as a measure of connectivity. The resulting connectivity matrices form a *convolutional network*. Let S^i_{jk} denote the total number of tracts between parcellations $\mathbf{R}^i_j$ and $\mathbf{R}^i_k$ at the ith layer (Figure 5.8). The connectivity S^i_{jk} at the ith layer is then the sum of connectivities at the $(i+1)$th layer (Figure 5.9), i.e.,

$$S^i_{jk} = \sum_{\mathbf{R}^{i+1}_l \subset \mathbf{R}^i_j} \sum_{\mathbf{R}^{i+1}_m \subset \mathbf{R}^i_k} S^{i+1}_{lm}.$$

The sum is taken over every subparcellation of $\mathbf{R}^i_j$ and $\mathbf{R}^i_k$. This provides a subject-level connectivity matrix. The connectivity matrix $S^i = (S^i_{jk})$ is expected to be very sparse at any hierarchy (Figure 5.9). Note the diagonal elements of matrix $S^i = (S^i_{jk})$, which defines node values, are unspecified yet. Then connectivity matrix S^i will also have a hierarchical structure.

Persistent homology may offer an effective framework in addressing the topological inconsistency in multiscale models. Instead of studying images and networks at a fixed scale, as usually is done in traditional approaches, persistent homology summarizes the changes of topological features over different scales and identifies the most persistent topological features that are robust under different scales. This robust performance under different scales is needed for network models that are parameter and scale dependent. Instead of building networks at one fixed parameter that may not be optimal, persistent homological approaches exploit the topological structure of the data

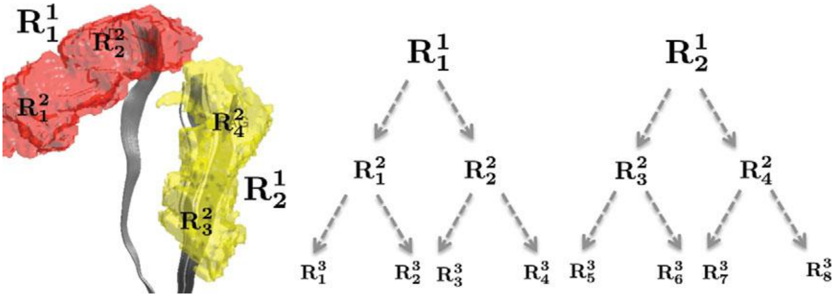


Figure 5.8 Two representative AAL parcellations $\mathbf{R}^1_1$ (right precentral gyrus) and $\mathbf{R}^1_2$ (left precentral gyrus) at the first layer will be partitioned into four subregions $\mathbf{R}^2_1, \mathbf{R}^2_2, \mathbf{R}^2_3$, and $\mathbf{R}^2_4$ at the second layer. The fiber tracts will be counted between the parcellations.

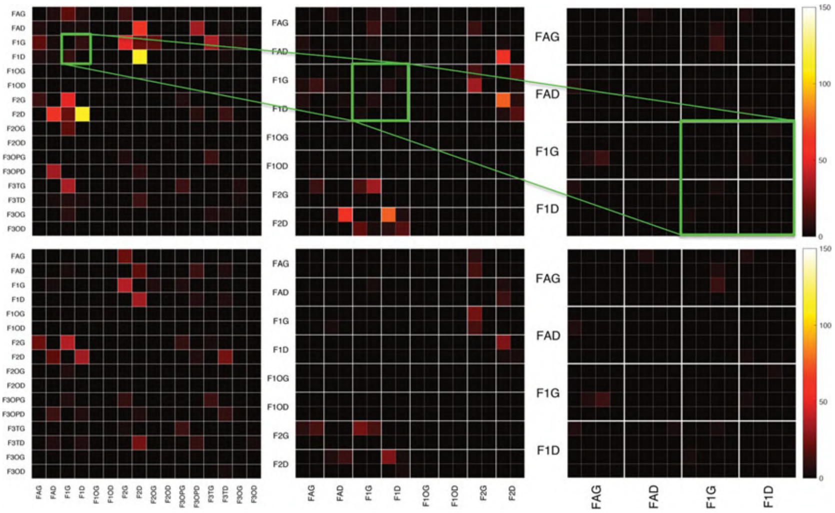


Figure 5.9 The hierarchical connectivity matrices of MZ- (top) and DZ-twins (bottom). The parts of connectivity matrices of the layers 1, 2, and 3 are shown. They form a layered convolutional network, where the convolution is defined as the sum of tracts between subparcellations.

and models. In doing so, topologically consistent nested hierarchical networks, called the *graph filtration*, are obtained (Lee et al., 2012; Chung et al., 2015a). Such a nested hierarchical structure can further speed up various computations even for large-scale networks with a billions of connections (Chung et al., 2017a).

5.4 Computing Large Correlation Matrices

For constructing large-scale correlation-based brain networks, a different type of correlation computation technique is needed. Existing built-in functions in MATLAB and R packages for computing correlation matrices are not necessarily optimized for large-scale data. For large correlation matrices of size $p \times p$, where $p > 10,000$, the matrix computation takes a long time and requires significant memory (see Figure 5.10). The built-in functions in MATLAB and R packages are often not written in a computationally efficient fashion. The pairwise computation of correlations is not efficient. A more efficient way is to scale and normalize the data matrice first and compute the correlation matrix as a matrix product (Chung et al., 2015a, 2017a).

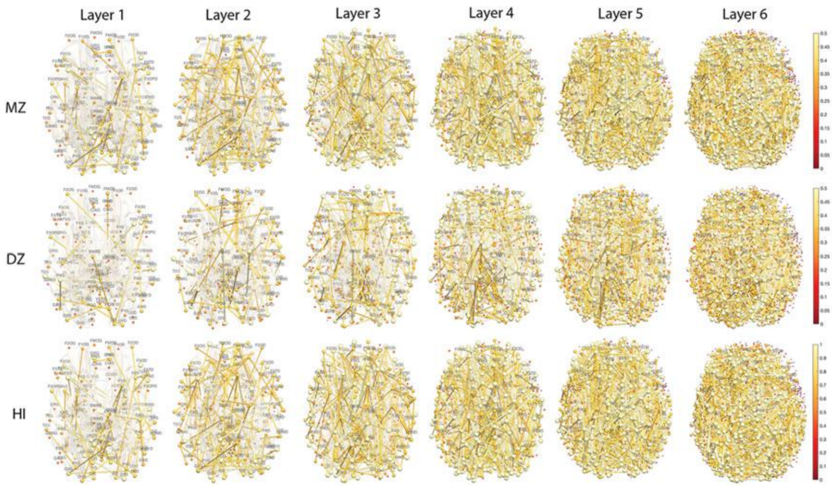


Figure 5.10 Top, middle: edge colors are Spearman's rank correlations thresholded at 0.3 for MZ- and DZ-twins for different layers. Node colors are the maximum correlation of all the connecting edges. Bottom: edge colors are the heritability index (HI). Node colors are the maximum HI of all the connecting edges. MZ-twins show higher correlations compared to DZ-twins. The node and edge sizes are proportionally scaled.

Suppose we have $n \times p$ data matrices $X = (x_{ij})$ and $Y = (x_{ij})$, where n is the number of images and p is the number of voxels we are interested in computing correlations. We scale and normalize along the columns such that

$$\sum_{i=1}^{n} x_{ij} = 0, \quad \sum_{i=1}^{n} x_{ij}^2 = 1, \tag{5.1}$$

$$\sum_{i=1}^{n} y_{ij} = 0, \quad \sum_{i=1}^{n} y_{ij}^2 = 1. \tag{5.2}$$

Then the correlation matrices of X and Y are defined as

$$corr(X) = X^\top X, \quad corr(Y) = Y^\top Y.$$

If $n \gg p$, the correlation matrices are full rank and invertible. If not, we no longer have well-defined correlation matrices and regularizations are possibly needed. The cross-correlation matrices of X and Y are defined as

$$corr(X,Y) = X^\top Y,$$

which is not symmetric. To make it symmetric, we can perform a symmetrization:

$$(X^\top Y + Y^\top X)/2.$$

The preceding procedure can be implemented as

```
function Z = corr2norm(X)
    [n p]=size(X);
    meanX=mean(X,1);
    meanX=repmat(meanX,n,1);
    X1=X-meanX;
    diagX1 = sqrt(diag(X1’*X1));
    X1=X1./repmat(diagX1’, n,1);
    Z=X1;
```

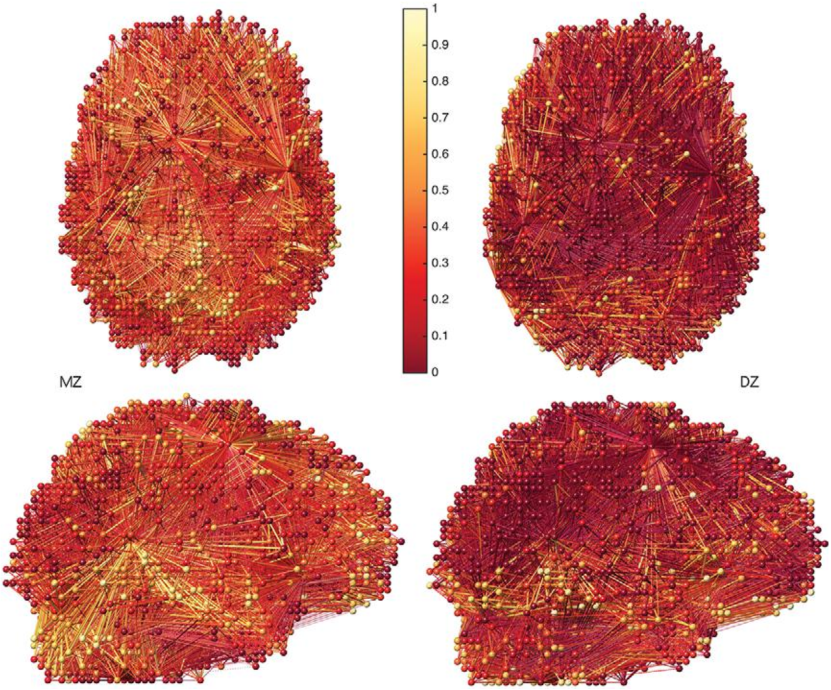


Figure 5.11 fMRI cross-correlation network of monozygotic (MZ) and same-sex dizygotic (DZ) twins with 25,972 nodes (Chung et al., 2017a). Only positive correlations are shown.

Once we normalize the data matrix, the cross-correlation matrix `C` is simply given as the product of martrices:

```
X= corr2norm(X);
Y = corr2norm(Y);
C = X'*Y;
```

This is perhaps the fastest way to compute large-scale correlation matrices. Due to scaling,

```
X'*X= 1
Y'*Y= 1
```

Often, data matrix `X` will likely to have missing values, which are often denoted as `NaN` in MATLAB. The preceding algorithm assumes there is no missing value. It may necessary to perform missing data treatment for any entry with `NaN`. Figure 5.11 displays the large-scale cross-correlation networks with 25,972 nodes constructed using the method (Chung et al., 2017a).

5.5 Online Algorithms

In terms of computation, many existing brain image analysis software such as SPM (`www.fil.ion.ucl.ac.uk/spm`) and AFNI (`afni.nimh.nih.gov`) are not effective for big data. The general statistical premise of such mainstream tools is that all the image measurements are available in the computer memory and statistics are computed using all the data. However, in the big data setting, it may not be possible to fit all of the imaging data in a computer's memory, making it necessary to perform the analysis by adding one image at a time in a sequential manner. We need a way to incrementally update the statistical analysis results without repeatedly running the entire analysis whenever new images or parts of images are added.

An *online algorithm* is one that processes its inputted data in a sequential manner (Chung et al., 2017c). Instead of processing the entire set of imaging data from the start, an online algorithm processes one image at a time. That way, we can bypass the memory requirement, reduce numerical instability, and increase computational efficiency. With the ever-increasing amount of large-scale brain imaging data sets such as ADNI and HCP, the development of various online statistical methods is warranted (Chung et al., 2017c). Thus, here is an immediate need to develop the online version of sparse or hierarchical network models, although there are no such available methods yet. Even large-scale Pearson correlation coefficients can be computed using an online algorithm.

Existing statistical analysis packages such as MATLAB and R also assume all measurements to be available in computer memory. Unless substantial modification to existing codes is made, we cannot even compute t-statistics for extremely large data that will not fit into the computer memory using the built-in functions. Thus, there is a strong need to develop online algorithms for big data beyond brain imaging.

5.5.1 Online Two-Sample t-Test

Given data $x_1, \cdots, x_m$, an *online algorithm* for computing the sample mean μ_m is given by

$$\begin{aligned}\mu_m &= \frac{1}{m}\sum_{i=1}^{m} x_i \\ &= \mu_{m-1} + \frac{1}{m}(x_m - \mu_{m-1})\end{aligned}$$

for any $m \geq 1$. The algorithm updates the previous mean μ_{m-1} with new data x_m. This algorithm avoids accumulating large sums and tend to be numerically more stable (Finch, 2009).

An online algorithm for computing the sample variance σ_m^2 is algebraically involved (Knuth, 1981; Chan et al., 1983). After lengthy derivation, it can be shown that

$$\begin{aligned}\sigma_m^2 &= \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \mu_m)^2 \\ &= \frac{m-2}{m-1}\sigma_{m-1}^2 + \frac{1}{m}(x_m - \mu_{m-1})^2\end{aligned}$$

for $m \geq 2$. The algorithm starts with the initial value $\sigma_1^2 = 0$.

For comparing a collection of data between groups, two-sample t-statistic can be used. Given measurements $x_1, \cdots, x_m \sim N(\mu^1, (\sigma^1)^2)$ in one group and $y_1, \cdots, y_n \sim N(\mu^1, (\sigma^2)^2)$ in the other group, the two-sample t-statistic for testing

$$H_0 : \mu^1(x) = \mu^2(x) \;\; vs. \;\; H_1 : \mu^1(x) > \mu^2(x)$$

at each x is given by

$$T_{m,n}(x) = \frac{\mu_m^1 - \mu_n^2 - (\mu^1 - \mu^2)}{\sqrt{(\sigma^1)_m^2/m + (\sigma^2)_n^2/n}},$$

where μ^1_m, μ^2_n, $(\sigma^1)^2_m, (\sigma^2)^2_m$ are sample means and variances in each group estimated using the online algorithm. $T_{m,n}$ is then sequentially computed as

$$T_{1,0} \rightarrow T_{2,0} \rightarrow \cdots \rightarrow T_{m,0} \rightarrow T_{m,1} \rightarrow \cdots \rightarrow T_{m,n}$$

in $m + n$ steps.

5.5.2 Online Algorithm for Linear Regression

The online algorithm for linear regression is itself useful but additionally more useful in constructing an online algorithm for F-tests in the next section. Given data vector $\mathbf{y}_{m-1} = (y_1, \cdots, y_{m-1})^\top$ and design matrix Z_{m-1}, consider linear model

$$\mathbf{y}_{m-1} = Z_{m-1}\boldsymbol{\lambda}_{m-1}$$

with unknown parameter vector $\boldsymbol{\lambda}_{m-1} = (\lambda_1, \lambda_2, \cdots, \lambda_k)^\top$. Z_{m-1} is a matrix of size $(m-1) \times k$. Multiplying $Z^\top_{m-1}$ on the both sides, we have

$$Z^\top_{m-1}\mathbf{y}_{m-1} = Z^\top_{m-1}Z_{m-1}\boldsymbol{\lambda}_{m-1}. \tag{5.3}$$

Let $W_{m-1} = Z^\top_{m-1}Z_{m-1}$, which is a $k \times k$ matrix. In most applications, there are substantially more data than the number of parameters, i.e., $m \gg k$, and W_{m-1} is invertible. The least squares estimation (LSE) of $\boldsymbol{\lambda}_{m-1}$ is given by

$$\boldsymbol{\lambda}_{m-1} = W^{-1}_{m-1}Z^\top_m\mathbf{y}_{m-1}.$$

When new data y_m are introduced to the linear model (5.3), the model is updated to

$$\begin{pmatrix} \mathbf{y}_{m-1} \\ y_m \end{pmatrix} = \begin{pmatrix} Z_{m-1} \\ z_m \end{pmatrix} \boldsymbol{\lambda}_m,$$

where z_m is a $1 \times k$ row vector. Subsequently, we have

$$(Z^\top_{m-1}\ z^\top_m) \begin{pmatrix} \mathbf{y}_{m-1} \\ y_m \end{pmatrix} = (Z^\top_{m-1}\ z^\top_m) \begin{pmatrix} Z_{m-1} \\ z_m \end{pmatrix} \boldsymbol{\lambda}_m$$

$$Z^\top_{m-1}\mathbf{y}_{m-1} + z^\top_m y_m = (W_{m-1} + z^\top_m z_m)\boldsymbol{\lambda}_m.$$

From Woodbury formula (Deng, 2011), we can write

$$(W_{m-1} + z^\top_m z_m)^{-1} = W^{-1}_{m-1} - c_m W^{-1}_{m-1}z^\top_m,$$

where $c_m = 1/(1 + z_m W_{m-1} z^\top_m)$ is scalar. Then we have the explicit online algorithm for updating the parameter vector:

$$\boldsymbol{\lambda}_m = (I - W^{-1}_{m-1}z^\top_m y_m - c_m W^{-1}_{m-1}z^\top_m W^\top_{m-1})\boldsymbol{\lambda}_{m-1} - c_m W^{-1}_{m-1}z^\top_m z^\top_m y_m,$$

where I is the identity matrix of size $k \times k$. Since the algorithm requires W_{m-1} to be invertible, the algorithm must start from

$$\boldsymbol{\lambda}_k \rightarrow \boldsymbol{\lambda}_{k+1} \rightarrow \cdots \rightarrow \boldsymbol{\lambda}_m.$$

At each iteration, we need to store $k \times k$ matrix W_{m-1}. In many applications, k will not be larger than 10 and most likely around 5 or less, which is manageable as far as computer memory is concerned.

A similar online algorithm for fitting a general linear model (GLM) was introduced for real-time fMRI (Bagarinao et al., 2006), where the Cholesky factorization was used to invert the covariance matrix in solving GLM. Our approach based on the Woodbury formula does not require the factorization or inversion of matrices.

5.5.3 Online Algorithm for *F*-Test

An online algorithm for the F-test is involved, but it is based on the online algorithm for linear regression. Let y_i be the ith data, $\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})^\top$ be the variables of interest, and $\mathbf{z}_i = (z_{i1}, \cdots, z_{ik})^\top$ be nuisance covariates corresponding to the ith data. We assume there are $m-1$ data to start with. Consider a general linear model (Chung, 2013)

$$\mathbf{y}_{m-1} = Z_{m-1}\boldsymbol{\lambda}_{m-1} + X_{m-1}\boldsymbol{\beta}_{m-1},$$

where $Z_{m-1} = (z_{ij})$ is $(m-1) \times k$ design matrix, $X_{m-1} = (x_{ij})$ is $(m-1) \times p$ design matrix. $\boldsymbol{\lambda}_{m-1} = (\lambda_1, \cdots, \lambda_k)^\top$ and $\boldsymbol{\beta}_{m-1} = (\beta_1, \cdots, \beta_p)^\top$ are unknown parameter vectors to be estimated at the $(m-1)$th iteration. Consider hypotheses

$$H_0 : \boldsymbol{\beta} = 0 \text{ vs. } H_1 : \boldsymbol{\beta} \neq 0.$$

The reduced null model when $\boldsymbol{\beta} = 0$ is

$$\mathbf{y}_{m-1} = Z_{m-1}\boldsymbol{\lambda}^0_{m-1}.$$

The goodness-of-fit of the null model is measured by the SSE:

$$\text{SSE}^0_{m-1} = (\mathbf{y}_{m-1} - Z_{m-1}\boldsymbol{\lambda}^0_{m-1})^\top(\mathbf{y}_{m-1} - Z_{m-1}\boldsymbol{\lambda}^0_{m-1}),$$

where $\boldsymbol{\lambda}^0_{m-1}$ is estimated using the online algorithm (5.4). This provides the sequential update of SSE under H_0:

$$\text{SSE}^0_k \rightarrow \text{SSE}^0_{k+1} \rightarrow \cdots \rightarrow \text{SSE}^0_m.$$

Similarly, the fit of the alternate full model is measured by

$$\text{SSE}^1_{m-1} = (\mathbf{y}_{m-1} - \mathbb{Z}_{m-1}\boldsymbol{\gamma}^1_{m-1})^\top(\mathbf{y}_{m-1} - \mathbb{Z}_{m-1}\boldsymbol{\gamma}^1_{m-1}),$$

where $\mathbb{Z}_{m-1} = [Z_{m-1}X_{m-1}]$ is the combined design matrix and of size $(m - 1) \times (k + p)$, and

$$\boldsymbol{\gamma}^1_{m-1} = \begin{pmatrix} \boldsymbol{\lambda}^1_{m-1} \\ \boldsymbol{\beta}^1_{m-1} \end{pmatrix}$$

is the combined parameter vector of size $(k+p) \times 1$. Similarly, using the online algorithm (5.4), SSE under H_1 is given as

$$\text{SSE}^1_{k+p} \to \text{SSE}^1_{k+1} \to \cdots \to \text{SSE}^1_m.$$

Under H_0, the test statistic at the mth iteration f_m is given by

$$f_m = \frac{(\text{SSE}_0 - \text{SSE}_1)/p}{\text{SSE}_0/(m - p - k)} \sim F_{p,m-p-k},$$

which is the F-statistic with p and $m - p - k$ degrees of freedom.

5.5.4 Online Algorithm for a Correlation Matrix

Using a similar iterative principle, it is possible to update the Pearson correlation when new data are added. Bivariate data $\mathbf{x}_n = (x_1, x_2, \cdots, x_n)$ and $\mathbf{y}_n = (y_1, y_2, \cdots, y_n)$ are given. The Pearson correlation between $\mathbf{x}_n$ and $\mathbf{y}_n$ is given by $\rho(\mathbf{x}_n, \mathbf{y}_n)$. Suppose new data x_{n+1} and y_{n+1} are added to existing data $\mathbf{x}_n$ and $\mathbf{y}_n$ respectively. Then it is possible to compute the Pearson correlation between $\mathbf{x}_{n+1} = (\mathbf{x}_n, x_{n+1})$ and $\mathbf{y}_{n+1} = (\mathbf{y}_n, y_{n+1})$ as a function of $\rho(\mathbf{x}_n, \mathbf{y}_n)$ and x_n and y_n only. The numerical implementation will input existing correlation value $\rho(\mathbf{x}_n, \mathbf{y}_n)$ and new data x_{n+1} and y_{n+1}, then outputs $\rho(\mathbf{x}_{n+1}, \mathbf{y}_{n+1})$. One possible iterative solution is[1]

$$n\mathbb{V}(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}) = (n - 1)\mathbb{V}(\mathbf{x}_n, \mathbf{x}_n) + \frac{n}{n+1}(\mathbf{x}_{n+1} - \mu_n)(\mathbf{y}_{n+1} - \mu_n),$$

where μ_n is the online algorithm for the sample mean and $\mathbb{V}$ is the sample covariance. The normalization of the sample covariance is needed to obtain the online version of correlation. It is possible to have slightly different variations to the preceding formulation.

We can further construct an online algorithm for computing the correlation matrix. Suppose n subjects are given. Each subject has p measurements. The ith subject data vector can be denoted as

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^\top.$$

[1] The formulation was derived by Tiankang Xie of University of Wisconsin–Madison and Zewei Lin of Renmin University.

The $p \times p$ correlation matrix $C_n = (c_{ij}^n)$ across subjects is given by $c_{ij}^n = \rho(\mathbf{x}_i, \mathbf{x}_j)$, the Pearson correlation between $\mathbf{x}_i$ and $\mathbf{x}_j$. Now we add the $(n+1)$th subject with p measurements, denoted as $\mathbf{x}_{n+1}$. Then the correlation matrix C_{n+1} of all $(n+1)$ subjects can be computed using only C_n and $\mathbf{x}_{n+1}$. The resulting $p \times p$ correlation matrix C_{n+1} is a function of C_n and $\mathbf{x}_{n+1}$ only and the online algorithm is based on the iterative formula. One possible solution can be derived by making the matrix version of the iterative algorithm for correlations.

6

Network Simulations

This chapter covers a practical issue often needed but ignored in brain network analysis. For validating and comparing different brain network models, it is often necessary to simulate networks with specific properties. In this chapter, we will study how to generate complex networks randomly using the mathematical ground truth (Chung et al., 2015a, 2017a,d). The basic tools for simulating complex random networks statistically are multivariate data analysis and mixed-effect models.

6.1 Multivariate Normal Distributions

We start with reviewing multivariate normal distributions. The multivariate normal distribution is a generalization of the univariate normal distribution to higher dimensions. So it can be defined from normal distributions as follows.

Definition 6.1 *Random variable Z has* standard normal *distribution, denoted as $N(0,1)$, if its probability density function is*

$$P(Z = z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}.$$

The cumulative distribution function *of Z is the probability*

$$P(Z \leq z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} \, dz. \tag{6.1}$$

If Z is standard normal, it is often denoted as

$$Z \sim N(0,1).$$

The integral (6.1) is often called the error function in engineering literature. All other normal distributions are a linear transformation of Z.

Definition 6.2 *If Z is standard normal, $X = \sigma Z + \mu$ is distributed as normal with mean μ and variance σ^2. This is denoted as $N(\mu, \sigma^2)$.*

Definition 6.3 *Given random vectors Z and W, the cross-covariance matrix is given by*

$$\mathbb{V}(Z, W) = \mathbb{E}\big[(Z - \mathbb{E}Z)(W - \mathbb{E}W)^\top\big].$$

The covariance matrix of Z is then defined as

$$\mathbb{V}Z = \mathbb{V}(Z, Z).$$

The algebraic derivation can show that (6.2) can be rewritten as

$$\mathbb{V}Z = \mathbb{E}(ZZ^\top) - \mathbb{E}Z(\mathbb{E}Z^\top).$$

Definition 6.4 *Consider random vector $Z = (z_1, \cdots, z_p)^\top$, where each component is independent and identically distributed as $z_i \sim N(0,1)$. Then for vector μ and matrix H, $X = \mu + HZ$ is distributed as* multivariate normal *with mean μ and the covariance matrix $HH^\top$. This is denoted as*

$$X \sim N(\mu, HH^\top).$$

The covariance matrix $HH^\top$ is computed as follows

$$\mathbb{V}(HZ) = \mathbb{E}\big[(HZ)(HZ)^\top\big] = HH^\top$$

since $\mathbb{E}(ZZ^\top) = I$, the identity matrix. Thus, the covariance matrix is symmetric positive definite.

6.1.1 Cholesky Factorization

Theorem 6.1 *(Cholesky factorization) Any $p \times p$ symmetric positive definite matrix $V = (v_{ij})$ can be factored as $V = HH^\top$, where $H = (h_{ij})$ is a lower triangular matrix with real and positive diagonal entries and $H^\top$ is the upper triangular matrix.*

The proof is given in Harville (1997).

Theorem 6.2 *Given the Cholesky factorization of SPD matrix $V = HH^\top$, $V = (v_{ij}), H = (h_{ij})$, we have*

$$h_{jj} = \Big(v_{jj} - \sum_{k=1}^{j-1} h_{jk}^2\Big)^{1/2},$$

$$h_{ij} = \frac{v_{ij} - \sum_{k=1}^{j-1} h_{ik}h_{jk}}{h_{jj}} \quad \textit{for } i > j.$$

The main advantage of Cholesky factorization is the explicit formulation Theorem 6.2 for factorization. The Cholesky factorization has been often used for various matrix computations as well as proving difficult theorems. The Cholesky factorization can be used to simulate random networks with specific covariance matrix as edge weights. MATLAB implements the Cholesky factorization as `chol.m`. The MATLAB convention uses the upper triangular matrix as the Cholesky factor while many textbooks use the lower triangular matrix. In MATLAB, the Cholesky factor $H^\top$ of $V = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ is computed as follows:

```
V=[2 1
     1 2];

Htr=chol(V)
Htr =
    1.4142    0.7071
         0    1.2247

Htr'*Htr
ans =
    2.0000    1.0000
    1.0000    2.0000
```

Consider simulating 10,000 multivariate normal random vectors with zero mean and covariance

$$V = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

can be done via Cholesky factorization as follows. We use the Gaussian random number generator `normrnd`, which generates a collection of $N(0, 1)$ variables:

```
z1 = normrnd(0,1,10000,1);
z2=normrnd(0,1,10000,1);
W=Htr'*[z1 z2]';
figure; plot(W(1,:), W(2,:),'.k')
```

The result is displayed in Figure 6.1. The covariance matrix is then estimated by computing the sample covariance:

```
cov(W')
ans =
```

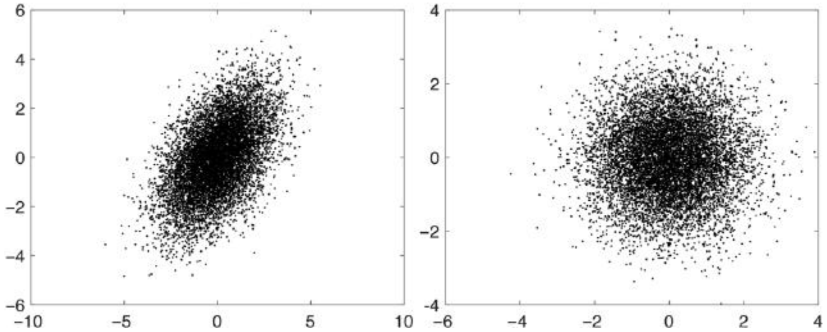


Figure 6.1 Left: 10,000 mean-zero bivariate normal vectors with covariance matrix V. The principal eigenvector of V corresponds to the direction of data elongation. Right: using the Cholesky factorization, we can uncorrelate the data.

```
    2.0475    1.0379
    1.0379    2.0381
```

Even with 10,000 data points, the covariance matrix is not easy to estimate accurately.

6.1.2 Gaussianness

Often it is necessary to check Gaussianness of network data. Since many statistical network models assume normality, checking if imaging data follows normality is an important problem. The normality of data is easily checked visually using the *quantile-quantile (QQ) plot* first introduced by (Wilk and Gnanadesikan, 1968). The QQ-plot is a visualization technique for comparing two distributions by plotting their quantiles against each other. A special case of QQ-plot is the *normal probability plot* where the quantiles from an empirical distribution of data are plotted on the vertical axis while the theoretical quantiles from a Gaussian distribution are plotted on the horizontal axis. It is used to check graphically if the empirical distribution follows the theoretical Gaussian distribution. If the data follow Gaussian, the normal probability plot should be close to a straight line.

Definition 6.5 *The* quantile point *q for random variable X is a point that satisfies*

$$P(X \leq q) = F_X(q) = p,$$

where F_X is called the cumulative distribution function *(CDF) of X.*

Since CDF is monotonically increasing, we can find the inverse of CDF. Then the quantile q is given by

$$q = F_X^{-1}(p).$$

This function is mainly referred to as the quantile function. The QQ-plot of two random variables X and Y is then defined to be a parametric curve $\mathcal{C}(p)$ parameterized by $p \in [0,1]$:

$$\mathcal{C}(p) = \big(F_X^{-1}(p), F_Y^{-1}(p)\big).$$

Theorem 6.3 *The QQ-plot* (6.2) *for two Gaussian distributions is a straight line.*

Proof. Suppose

$$X \sim N(\mu_1, \sigma_1^2), \quad Y \sim N(\mu_2, \sigma_2^2).$$

Let $Z \sim N(0,1)$ and $\Phi(z) = P(Z \le z)$, the CDF of the standard normal distribution. Denote q_1 and q_2 to be the pth quantiles for X and Y respectively. Then we have

$$\begin{aligned} p &= P(X \le q_1) \\ &= P\Big(\frac{X - \mu_1}{\sigma_1} \le \frac{q_1 - \mu_1}{\sigma_1}\Big) \\ &= \Phi\Big(\frac{q_1 - \mu_1}{\sigma_1}\Big). \end{aligned}$$

Hence the parameterized QQ-plot is given by

$$\begin{aligned} q_1(p) &= \mu_1 + \sigma_1 \Phi^{-1}(p), \\ q_2(p) &= \mu_2 + \sigma_2 \Phi^{-1}(p). \end{aligned}$$

This is an explicit parametric form of the QQ-plot. The implicit form without $\Phi^{-1}(p)$ is given by

$$\frac{q_1 - \mu_1}{\sigma_1} = \frac{q_2 - \mu_2}{\sigma_2},$$

the equation for a line. This shows the QQ-plot of two normal distributions is a straight line. The ratio of variability σ_1/σ_2 determines the slope of the line. □

Theorem 6.3 can be used to determine the normality of scalar data. We can check how closely the sample quantiles correspond to the normal distribution by plotting the QQ-plot of the sample quantiles vs. the corresponding quantiles of a normal distribution. In normal probability plot, we plot the QQ-plot of the sample against the standard normal distribution $N(0,1)$. The sample quantiles are obtained using the empirical CDF.

6.1.3 Empirical Distributions

The CDF $F_X(q)$ measures the proportion of random variable X less than given value q. So by counting the number of measurements less than q, we can empirically estimate the CDF. Let $X_1, \cdots, X_n$ be a random sample of size n. Then order them in increasing order:

$$\min(X_1, \cdots, X_n) = X_{(1)} \le X_{(2)} \le \cdots \tag{6.2}$$

$$\le X_{(n)} = \max(X_1, \cdots, X_n). \tag{6.3}$$

The monotonic sequence of random variables $X_{(1)}, \cdots, X_{(n)}$ is often called the order statistic. Suppose $X_{(j)} \le q < X_{(j+1)}$. This implies that there are j samples that are smaller than q. So we approximate the CDF as

$$\widehat{F_X}(q) = \frac{j}{n}.$$

The j/nth *sample quantile* is then $X_{(j)}$ (Chung, 2013). Some authors define the sample quantile as the $(j - 0.5)/n$th sample quantile. The factor 0.5 is introduced to account for the descritization error.

In numerical implementation, it is easier to implement the empirical distribution using the *step function* $\mathcal{I}_q(x)$ which is implemented as

$$\mathcal{I}_q(x) = \begin{cases} 1 & \text{if } x \le q \\ 0 & \text{if } x > q \end{cases}.$$

Then the CDF is empirically estimated as

$$\widehat{F_X}(q) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{I}_q(X_i),$$

where $\mathcal{I}_q(X_i)$ counts how many times X_i is less than q. A different, possibly more sophisticated CDF estimation can be found in (Frigge et al., 1989).

Example 6.1 *Consider the problem of plotting the quantile function for the exponential random variable X with parameter $\lambda = 2$. Its probability density function is given by*

$$f(x) = \lambda e^{-\lambda x}.$$

It can be shown that

$$F_X^{-1}(p) = -\frac{1}{2}\ln(1 - p).$$

We generate the QQ-plot of $X \sim exp(2)$ vs. $Y \sim exp(2)$ using the exponential random number generator `exprnd` *(Figure 6.2). Since they are identical distributions, we expect the straight line as the QQ-plot:*

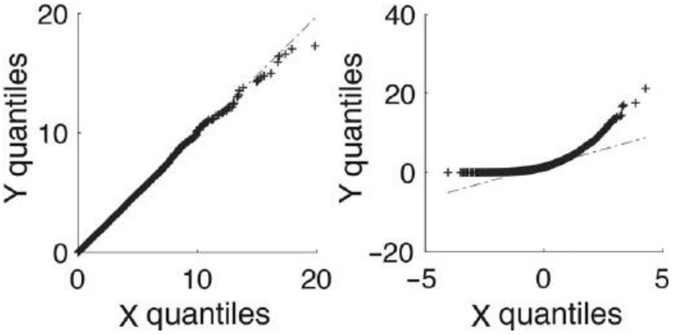


Figure 6.2 Left: QQ-plot of $X \sim exp(2)$ over $Y \sim exp(2)$. We expect the straight line. Right: QQ-plot of $X \sim N(0, 1)$ over $Y \sim exp(2)$.

```
X=exprnd(2, 10000,1);
Y=exprnd(2, 10000,1);
subplot(1,2,1); qqplot(X,Y);
```

When we generate the QQ-plot of $X \sim N(0, 1)$ vs. $Y \sim exp(2)$, we get a curve (Figure 6.2):

```
X=normrnd(0,1, 10000,1);
Y=exprnd(2, 10000,1);
subplot(1,2,2); qqplot(X,Y);
```

6.1.4 Multivariate Normality

Often it is necessary to check multivariate Gaussian assumptions in practice. Given multivariate vector W, we are interested in testing if the data follow variate normal. We can use the χ^2 goodness-of-fit test, but a simpler approach would be to check if given bivariate data can be generated from normal distributions:

$$W = HZ + \mu, Z \sim N(0, I)$$

for some unknown H and μ. Thus, we need to check if

$$H^{-1}(W - \mu) = Z \sim N(0, I).$$

Example 6.2 *Let us check if W is distributed as a bivariate Gaussian. In this particular example, we assumed the mean of data is zero, so there is no need to estimate μ.*

```
Htr = Htr'
    1.4142         0
    0.7071    1.2247

inv(Htr)
    0.7071         0
   -0.4082    0.8165

Z=inv(Htr)*W;
figure; plot(Z(1,:), Z(2,:) ,'.k')
```

The components of Z become uncorrelated (Figure 6.1). Then we check the normality of each component of Z using quantile-quantile plots.

6.2 Multivariate Linear Models

Consider a simple higher-dimensional generalization of linear model:

$$u(x) = \mu(x) + \Sigma^{1/2}(x)\epsilon(x), \tag{6.4}$$

where u is a vector of observations at position x, μ is the unknown mean vector and $\Sigma(x)$ is the symmetric positive-definite covariance matrix, which allows for correlations between components of u and depends on the coordinates x only (Worsley et al., 1996b; Cao and Worsley, 1999a; Chung et al., 2001b). Since Σ is symmetric positive-definite, the square root of Σ always exists. The components of the error vector ϵ are assumed to be independent and identically distributed as smooth stationary Gaussian random fields with zero mean and unit standard deviation. The model has been widely used in brain imaging (Worsley, 1994; Worsley et al., 1996b, 2004; Thompson et al., 1997; Collins et al., 1998; Joshi, 1998; Gaser et al., 1999; Cao and Worsley, 1999a; Chung et al., 2001b; Chung, 2013; Worsley et al., 1996b, 2004).

6.2.1 Hotelling's T-Square Statistic

We are interested in detecting testing the statistical significant in linear model (6.4). This is a standard multivariate statistical inference problem and can be solved using Hotelling's T^2 statistic (Worsley, 1994; Worsley et al., 1996b, 2004; Thompson et al., 1997; Collins et al., 1998; Joshi, 1998; Cao and Worsley, 1999a; Gaser et al., 1999; Chung et al., 2001b). Hotelling's T^2 statistic is fairly flexible and can be applicable to wide variety of situations.

If we only have one group, we can simply assume μ_2 to be a known vector field and treat the problem as a one-sample problem. In the case of a longitudinal study, where two scans per subject are available, we can take μ_1 as the growth velocity by dividing the displacement difference by the scan interval. Then we are testing if there is any significant growth over time.

Under the assumption (6.4), we are interested in testing if the two groups have the same vector mean:

$$\begin{aligned} H_0 &: \mu_1(x) = \mu_2(x) \text{ for all } x \\ &\quad \text{vs.} \\ H_1 &: \mu_1(x) \neq \mu_2(x) \text{ for some } x, \end{aligned}$$

where μ_i is the unknown d-dimensional mean vector field for the ith group. The inference is based on Hotelling's T^2-statistic. Let us rewrite (6.4) for an individual subject using the group index i and the subject index j:

$$u^{ij}(x) = \mu^i(x) + \Sigma^{1/2}(x)\epsilon^{ij}(x),$$

where μ^{ij} are the ith group mean vector and ϵ^{ij} are independent and identically distributed Gaussian random vector field. Let n_i be the number of subjects in the ith group. The unknown ith group mean μ^i is estimated as

$$\overline{\mu}^i = \frac{1}{n_i}\sum_{j=1}^{n_i} u^{ij}.$$

Testing the hypotheses is then done by checking the significance of the mean difference $\overline{\mu}^2 - \overline{\mu}^1$. The pooled sample covariance matrix is given by

$$\widehat{\Sigma} = \frac{1}{n_1+n_2-2}\Big[\sum_{j=1}^{n_1}(u^{1j} - \overline{\mu}^1)(u^{1j} - \overline{\mu}^1)^\top + \sum_{j=1}^{n_2}(u^{2j} - \overline{\mu}^2)(u^{2j} - \overline{\mu}^2)^\top\Big].$$

The significance of the group difference is then tested using the Hotelling's T^2-statistic

$$H(x) = \frac{n_1 n_2(n_1+n_2-d-1)}{d(n_1+n_2)(n_1+n_2-2)}(\overline{\mu}^2 - \overline{\mu}^1)^\top \widehat{\Sigma}^{-1}(\overline{\mu}^2 - \overline{\mu}^1).$$

At each x, under the null hypothesis of $\mu_1(x) = \mu_2(x)$, H is distributed as an F-statistic with d and $n_1 + n_2 - d - 1$ degrees of freedom. This is for two samples, but a one-sample case is similar (Chung et al., 2001b).

For the one-sample case, we assume μ_2 is a known constant vector field and the corresponding Hotelling's T^2 statistic is given by

$$H(x) = \frac{n_1(n_1-d)}{d(n_1-1)}(\overline{\mu}^1 - \mu_2)^\top \widehat{\Sigma}^{-1}(\overline{\mu}^1 - \mu_2), \tag{6.5}$$

where the sample covariance is given by

$$\widehat{\Sigma} = \frac{1}{n_1 - 1} \sum_{j=1}^{n_1} (u^{1j} - \overline{\mu}^1)(u^{1j} - \overline{\mu}^1)^\top.$$

At each voxel x, under the hypothesis, $H(x)$ is distributed as an F-statistic with d and $n_1 - d$ degrees of freedom.The following 2D example illustrates how to perform Hotelling's T^2 in MATLAB.

Example 6.3 *We are interested in testing the equality of 2D vector measurements in two groups. The measurements for the first group are*

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ 3 \end{pmatrix}.$$

The measurements for the second group are

$$\begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

The data are coded as

```
mu1= [0 1; 0 2; 0 3]
mu2=[2 0; 0 0]
n1=3; n2=2; d=2;
```

where `n1` and `n2` are the sample sizes and `d` is the dimension of the vector.

The sample means for the two groups and the mean difference are

$$\bar{\mu}^1 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \bar{\mu}^2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \ \bar{\mu}^1 - \bar{\mu}^2 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}.$$

The pooled variance is estimated as

$$\widehat{\Sigma} = \frac{1}{3} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

This is computed as

```
m1 = mean(mu1,1)
m2=mean(mu2,1)

z1=mu1-kron(ones(n1,1),m1);
z2=mu2-kron(ones(n2,1),m2);

sigma= (z1'*z1 + z2'*z2)/(n1+n2-2)
```

```
sigma =

    0.6667         0
         0    0.6667
```

The inverse covariance matrix is then trivially

$$\widehat{\Sigma}^{-1} = \frac{3}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The Hotelling's T-square statistic value is then given as

$$h = \frac{3 \cdot 2(3+2-2-1)}{2(3+2)(3+2-2)}(\bar{\mu}^2 - \bar{\mu}^1)^\top \widehat{\Sigma}^{-1}(\bar{\mu}^2 - \bar{\mu}^1) = 3.$$

The Hotelling's T-square is distributed as an F-statistic with 2 and 2 degrees of freedom. The p-value is given by $P(H > h) = 0.25$, which is computed as `1 - fcdf(3,2,2)`. Since the p-value is large, the group difference is not significant. The whole procedure is illustrated in MATLAB.[1]

6.2.2 Linear Discriminant Analysis

The *standard distance* with respect to distribution X is defined as

$$\Delta_X(x_1, x_2) = \frac{|x_1 - x_2|}{\sigma},$$

where $\sigma = \text{Var}(X)^{1/2}$. We will write $\Delta(x) = \Delta_X(x, \mu)$. Let $X = (X_1, \cdots X_p)^\top$ be a p-variate vector with

$$\mathbb{E}X = \mu \quad \text{and} \quad \mathbb{E}X = \Sigma.$$

Let $Y = a^\top X$ for some vector $a \in \mathbb{R}^p$. Then

$$\Delta_Y(y_1, y_2) = \frac{a^\top(x_1 - x_2)}{(a^\top \Sigma a)^{1/2})}.$$

This distance depends on the choice of vector a. So we define the p-variate standard distance, which is often called Mahalanobis distance, between two vectors with respect to X as

$$\Delta_X(x_1, x_2) = \max_{a \in \mathbb{R}^p, a \neq 0} \frac{a'(x_1, x_2)}{(a' \Sigma a)^{1/2}}.$$

[1] http://www.stat.wisc.edu/~mchung/research/amygdala/

This definition is scale invariant with respect to X. From the Cauchy–Schwartz inequality, we can show that

$$\Delta_X(x_1,x_2) = [(x_1 - x_2)^\top \Sigma^{-1}(x_1 - x_2)]^{1/2}.$$

For the ith population, $\mathbb{E}X = \mu_i$ while $\text{Cov}X = \Sigma$ is fixed, the multivariate distance between μ_1 and μ_2 is

$$\Delta_X(\mu_1,\mu_2) = [(\mu_1 - \mu_2)\Sigma^{-1}(\mu_1 - \mu_2)]^{1/2}.$$

$Y = \beta^\top X$ is a linear discriminant function for the two populations if

$$\Delta_Y(\beta^\top\mu_1,\beta^\top\mu_2) = \Delta_X(\mu_1,\mu_2).$$

$\beta = c \cdot \Sigma^{-1}(\mu_1 - \mu_2)$ satisfies this condition. For the choice of $c = 1$,

$$\text{var}(\beta^\top X) = \Delta_X^2(\mu_1,\mu_2) = \beta^\top(\mu_1 - \mu_2).$$

So for $c = 1$, the variance and the mean difference are identical.

We have the ith sample $x_{i1}, \cdots, x_{in}$. Define the sample multivariate standard distance between the sample means $\bar{x}_1$ and $\bar{x}_2$ as $D(\bar{x}_1,\bar{x}_2) = \Delta_X(\bar{x}_1,\bar{x}_2)$. Then it is given by

$$D(\bar{x}_1,\bar{x}_2) = [(\bar{x}_1 - \bar{x}_2)S^{-1}(\bar{x}_1 - \bar{x}_2)]^{1/2}$$

where S is the pooled sample covariance given by

$$S = \frac{n_1S_1 + n_2S_2}{n_1 + n_2 - 2}$$

where

$$S_j = \frac{1}{n_j - 1}\sum_{i=1}^{n_j}(x_{ji} - \bar{x}_j)(x_{ji} - \bar{x}_j)^\top.$$

Then the vector coefficients of the linear discriminant function are given by

$$\beta = S^{-1}(\bar{x}_1 - \bar{x}_2).$$

We classify the groups by the line $\beta X_1 + \beta X_2 = \text{const}$.

6.2.3 Covariates in Multivariate Linear Models

For a single contrast in the exploratory variables, Hotelling's T^2 can be used, but for multiple contrasts, it is natural to set up a multivariate general linear model (MGLM) and construct a test statistic (Worsley et al., 2004). Consider the following MGLM at each fixed point in the template:

$$P_{n\times q} = X_{n\times p}B_{p\times q} + Z_{n\times r}G_{r\times q} + U_{n\times q}\Sigma_{q\times q}, \tag{6.6}$$

where P is the matrix of response vectors, X is the matrix of contrasted explanatory variables, and B is the matrix of unknown coefficients. Nuisance covariates are in the matrix Z and the corresponding coefficients are in the matrix G. The subscripts denote the dimension of matrices. The components of Gaussian random matrix U are zero mean and unit variance. Σ accounts for the covariance structure of between q variables. Then we are interested in testing the null hypothesis

$$H_0 : B = 0 \text{ vs. } H_1 : B \neq 0.$$

For the reduced model corresponding to $B = 0$, the least squares estimator of G is given by solving $P = ZG$, i.e.,

$$\widehat{G}_0 = (Z^\top Z)^{-1} Z^\top P.$$

We will assume that there is more sample size n than the number of parameters r to be estimated. The residual sum of squares of the reduced model is

$$E_0 = (P - Z\widehat{G}_0)^\top (P - Z\widehat{G}_0).$$

For the full model, the parameters are estimated by solving

$$P_{n\times q} = XB + ZG = [X, Z]_{n\times(p+r)} \begin{bmatrix} B \\ G \end{bmatrix}_{(p+r)\times q}.$$

The least squares estimation is given by

$$\begin{bmatrix} \widehat{B} \\ \widehat{G} \end{bmatrix} = ([X, Z]^\top [X, Z])^{-1} [X, Z]^\top P.$$

The corresponding residual sum of squared error is

$$E = (P - X\widehat{B} - Z\widehat{G})^\top (P - X\widehat{B} - Z\widehat{G}).$$

By comparing how large the residual E is against the residual E_0, we can determine the significance of coefficients B. However, since E and E_0 are matrices, we take a function of eigenvalues of $E_0 E^{-1}$ as a statistic. Since we expect the sample size n to be larger than q, there are q eigenvalues

$$\lambda_1, \lambda_2, \cdots, \lambda_q$$

satisfying

$$\det(E_0 - \lambda E) = 0.$$

This requires solving the generalized eigenvalue problem

$$E_0 v = \lambda E v$$

for eigenvectors v. The q eigenvectors give the orthogonal linear combinations of the responses that produce maximal univariate F statistics (Fox et al., 2009). For instance, the Lawley–Hotelling trace is given by the sum of eigenvalues

$$\lambda_1 + \lambda_2 + \cdots + \lambda_q.$$

Wilks's Lambda is given by

$$(1+\lambda_1)^{-1}(1+\lambda_2)^{-1}\cdots(1+\lambda_q)^{-1}.$$

Roy's maximum root statistic R is the largest eigenvalue $\max_j \lambda_j$. The distributions of these multivariate test statistics are approximately F. In this case, there is only one eigenvalue, and all these multivariate test statistics simplify to Hotelling's T^2 statistic. Hotelling's T^2 statistic has been widely used in modeling 3D coordinates and deformations in brain imaging (Thompson et al., 1997; Joshi, 1998; Cao and Worsley, 1999a; Gaser et al., 1999; Chung et al., 2001b). The random field theory for Hotelling's T^2 statistic has been available for a while (Cao and Worsley, 1999a). However, the random field theory for the Roy's maximum root was not developed until recently (Worsley et al., 2004; Taylor and Worsley, 2008).

The inference for Roy's maximum root is based on the Roy's union-intersection principle (Roy, 1953; Worsley et al., 2004), which simplifies the multivariate problem to a univariate linear model. Let us multiply an arbitrary constant vector $\nu_{3\times 1}$ on both sides of (9.1):

$$P\nu = XB\nu + ZG\nu + U\Sigma\nu. \tag{6.7}$$

Obviously (6.7) is a usual univariate linear model with a Gaussian noise. For the univariate testing on $B\nu = 0$, the inference is based on the usual F statistic with p and $n-p-r$ degrees of freedom, denoted as F_ν. Then Roy's maximum root statistic is given by

$$R = \max_\nu F_\nu.$$

Now it is obvious that the usual random field theory can be applied in correcting for multiple comparisons. The only trick is to increase the search space, in which we take the supreme of the F random field, from the template surface to a much higher dimension to account for maximizing over ν as well. Another way of defining Roy's maximum root is via maximal canonical correlations (Worsley et al., 2004).

Keith Worsley's `SurfStat`[2] package has a built-in MATLAB routine for determining the p-value for Roy's maximum root statistic. `SurfStat`

[2] www.math.mcgill.ca/keith/surfstat

was developed to utilize a model formula and avoids the explicit use of design matrices and contrasts. SurfStat can import Montreal Neurological Institute's (MNI) (MacDonald et al., 2000), FreeSurfer-based[3] cortical mesh formats as well as other volumetric image data. A similar model formula approach is implemented in many other statistics packages such as Splus and Rand SAS. These statistics packages accept a linear model like

$$\texttt{Y} = \texttt{Group} + \texttt{Age} + \texttt{Brain}$$

as the direct input for linear modeling, avoiding the need to explicitly state the design matrix. Here, Y is a $n \times d$ matrix of measurements, Age is the age of subjects, Brain is the total brain volume of subject, and Group is the categorical group variable. This type of model formula has yet to be implemented in widely used SPM or AFNI packages.

6.3 Mixed Effects Models

Images in longitudinal and twin imaging studies are statistically dependent with high correlation. Individual networks are likely to be highly correlated within the same subject or across siblings. Thus, there is a need to model such dependent networks with explicit statistical models. The most obvious modeling choice is to use the *random effects model*, which is also called a variance components model, a special case of the hierarchical linear model. In statistical literature, fixed and random effects respectively refer to the population-average and subject-specific effects, which are often assumed to be unknown, i.e., latent variables. When a model has both random and fixed effects terms, the model is called *mixed effects*.

We will explain the mixed effects model using twin imaging study as an example. However, the method can easily extended to other dependent data settings such as longitudinal or multimodal images. Suppose the imaging data set consists of singles and twins. For the ith twin or single, consider the following mixed effects model (Milliken and Edland, 2000; Fox, 2002; Worsley et al., 2009)

$$y_i = X_i\beta + Z_i\gamma_i + \epsilon_i, \tag{6.8}$$

where y_i is $n_i \times 1$ vector of responses for the ith twin. n_i is either 1 (single) or 2 (twin). X_i is $n_i \times p$ design matrix corresponding to the fixed effects for the ith subject. Behavior and categorial dummy variables indicating twinness

[3] http://surfer.nmr.mgh.harvard.edu

can be treated as fixed effects. We will assume there is only one random effects term that varies across twins. The random effects term γ_i is of size $n_i \times 1$ and modeled as

$$\gamma_i \sim N(0, g_i),$$

where the covariance matrix g_i is of size $n_i \times n_i$. For singles, i.e., $n_i = 1$, $\gamma_i \sim N(0, \sigma^2)$, where σ^2 is the subject-level variability. For twins, i.e., $n_i = 2$,

$$\gamma_i \sim N\Big(0, \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\Big),$$

where ρ is the pairwise twin correlation. Thus, g_i is either σ^2 or $\sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ depending on if the ith cluster is a single or twin.

The model noise ϵ_i is assumed to follow

$$\epsilon_i \sim N(0, \sigma_\epsilon^2 I_{n_i}),$$

where I_{n_i} is the $n_i \times n_i$ identity matrix. The covariance of γ_i and ϵ_i are expected to have block diagonal structure such that there is no correlation among the different twins while there is high correlation within twins:

$$\mathbb{V}\begin{pmatrix} \gamma_i \\ \epsilon_i \end{pmatrix} = \begin{pmatrix} g_i & 0 \\ 0 & \sigma_\epsilon^2 I_{n_i} \end{pmatrix}.$$

The variance of y_i, denoted as V_i is given by

$$V_i = \mathbb{V} y_i = Z_i g_i Z i^\top + \sigma_\epsilon^2 I_{n_i}.$$

6.3.1 Matrix Form

In the numerical implementation, it is easier to use the matrix form. Assume there are total n twin pairs and singles so that the total sample size is

$$N = \sum_{i=1}^{n} n_i.$$

By combining (6.8), we have matrix form

$$Y = X\beta + Z\gamma + \epsilon, \tag{6.9}$$

where $Y = (y_1^\top, y_2^\top, \cdots, y_n^\top)^\top$ is the image measurement vector of every subject.

$$X = (X_1^\top, X_2^\top, \cdots, X_n^\top)^\top,$$
$$Z = diag(Z_1, Z_2, \cdots, Z_n)$$

are the design matrices corresponding to the fixed and random effects respectively. β is the fixed effects and $\gamma = (\gamma_1^\top, \gamma_2^\top, \cdots, \gamma_n^\top)^\top$ is the random effects. The combined noise vector

$$\epsilon = (\epsilon_1^\top, \epsilon_2^\top, \cdots, \epsilon_n^\top)^\top \sim N(0, R).$$

Note $R = \sigma_\epsilon^2 I_N$. The random effects follows $\gamma \sim N(0, G)$, where $G = diag(g_1, g_2, \cdots, g_n)$ accounts for covariance among random effect terms. Hierarchically, we are modeling (6.9) as

$$Y|\gamma \sim N(X\beta + Z\gamma, R),\ \gamma \sim N(0, G).$$

The covariance of Y, denoted as V, is then given by

$$V = \mathbb{V}Y = ZGZ^\top + R.$$

The parameters β, G, R are often estimated by maximizing the likelihood, which is computationally very demanding for most brain imaging studies. To speed up the computation, specific assumptions are often made into the structures of G and R. Note V is of size $N \times N$ block diagonal matrix consisting of $V_1, V_2, \cdots, V_n$ as the diagonal blocks.

6.3.2 Likelihood Methods

The fixed effects parameters β and random effect parameters $\alpha = (\alpha_1, \alpha_2, \alpha_3) = (\sigma^2, \sigma^2\rho, \sigma_\epsilon^2)$ are usually estimated by maximizing the likelihood function, although other methods are available. The likelihood methods are often computationally slow and not going to be easily scable for large-scale brain networks.

Let $f(y_i|\gamma_i)$ and $f(\gamma_i)$ be density functions for $y_i|\gamma_i$ and γ_i. The marginal density of y_i is then given by

$$y_i \sim N(X_i\beta, V_i).$$

Note the covariance V_i is a function of α only. Let

$$e_i(\beta) = y_i - X_i\beta$$

be the residual vector. The the likelihood function can be written as

$$L(\alpha, \beta) \propto \prod_{i=1}^{n} |V_i|^{-1/2} exp\left(-\frac{1}{2} e_i^\top V_i^{-1} e_i\right). \tag{6.10}$$

The log-likelihood is then

$$-l(\alpha,\beta) = -\log L(\alpha,\beta) \propto \sum_{i=1}^{n} \log|V_i| + e_i^\top V_i^{-1} e_i,$$

where $|V_i|$ denotes the determinant. Since $|V| = \prod_{i=1}^{n} |V_i|$, the combined matrix form is then

$$-l(\alpha,\beta) \propto \log|V(\alpha)| + e^\top(\beta) V^{-1}(\alpha) e(\beta)$$

with $e = (e_1, \cdots, e_n)^\top$. For fixed α, the weighted least squares estimate

$$\begin{aligned}\widehat{\beta}(\alpha) &= \left(\sum_{i=1}^{n} X_i V_i^{-1} Xi^\top\right)^{-1} \sum_{i=1}^{n} X_i V_i^{-1} y_i \\ &= (X^\top V^{-1} X)^{-1} X^\top V^{-1} Y \qquad (6.11)\end{aligned}$$

maximizes the likelihood (6.10). Note

$$\mathbb{E}\,\widehat{\beta}(\alpha) = \beta$$

and

$$\mathbb{V}\,\widehat{\beta}(\alpha) = \left(\sum_{i=1}^{n} X_i V_i^{-1} Xi^\top\right)^{-1} = (X^\top V^{-1} X)^{-1}.$$

6.3.3 Restricted Maximum Likelihood (REML)

Maximizing the likelihood without β produces the *restricted maximum likelihood* (REML) estimate for covariance parameters α (Fox, 2002; Pinehiro and Bates, 2002; Stroup, 2012). Consider any full rank matrix K satisfying $K^\top X = 0$. Then the marginal distribution of $K^\top Y$ does not depend on β since

$$K^\top Y \sim N(0, K^\top V K).$$

Thus we use the REML of $K'Y$ instead of the likelihood of Y. The residual operator $K = K^\top = I - XX^-$ can be used, where the pseudo-inverse of matrix A is given by $A^- = (A^\top A)^{-1} A'$. Note $KX = X - X^\top = 0$. Once $V(\alpha)$ is estimated by REML, β is subsequently estimated by plugging $\widehat{V}$, which gives the weighted least squares estimate. The REML l_{RE} has an extra term

$$-l_{RE}(\alpha,\beta) \propto \log|V| + e^\top V^{-1} e + \log|X^\top V^{-1} X|.$$

6.3.4 Numerical Implementation

We present a three-step method for estimating the parameters. This is the method similar to Keith Worsley's implemnetation of the REML procedure in `SurfStat` package [4] (Worsley et al., 2009; Chung, 2012).

(1) Fixed effects. We first start with estimating the fixed effects model parameters β and σ_ϵ^2 with $\sigma = 0$. This can be given by the weighted least squares estimation. However, if the stability is an issue, we may use the Cholesky factorization. Let U be the upper triangle Cholesky factor of covariance matrix V, i.e., $U^\top U = V$. Let $V_h = (U^\top)^{-1}$ be the inverse of lower triangle Cholesky factor. Note $V_h^\top V_h = V^{-1}$. Then (6.11) can be written as

$$\widehat{\beta} = (V_h X)^- V_h Y.$$

(2) Fisher scoring. Then the residual $Y - X\widehat{\beta}$ is used as response in estimating the variance parameters using REML. Then Fisher scoring will be used to iteratively estimate the parameters α (Stroup, 2012):

$$\alpha^{(j+1)} = \alpha^{(j)} + \frac{1}{\mathcal{I}_{RE}(\alpha^{(j)}, \beta)} \frac{\partial l_{RE}(\alpha, \beta)}{\partial \alpha}\Big|_{\alpha=\alpha^{(j)}}.$$

The initial value $\alpha^{(1)} = (0, 0, \sigma_\epsilon^2)$ is given from the fixed effects model, i.e. $\sigma^2 = 0$.

The gradient of the log-likelihood along the direction of α is

$$2\frac{\partial l(\alpha, \beta)}{\partial \alpha_i} = -\mathrm{tr}\Big(V^{-1}\frac{\partial V}{\partial \alpha_i}\Big) + e^\top V^{-1}\frac{\partial V}{\partial \alpha_i} V^{-1} e.$$

Similarly, the gradient of REML along the direction of α is

$$2\frac{\partial l_{RE}(\alpha, \beta)}{\partial \alpha_i} = -\mathrm{tr}\Big(P\frac{\partial V}{\partial \alpha_i}\Big) + e^\top V^{-1}\frac{\partial V}{\partial \alpha_i} V^{-1} e.$$

The information matrix of log-likelihood is given by

$$\mathcal{I}_{ij}(\alpha, \beta) = -\mathbb{E}\Big[\frac{\partial^2 l(\alpha, \beta)}{\partial \alpha_i \partial \alpha_j}\Big] = \frac{1}{2}\mathrm{tr}\Big(V^{-1}\frac{\partial V}{\partial \alpha_i} V^{-1}\frac{\partial V}{\partial \alpha_j}\Big),$$

where $P = V^{-1} - V^{-1}X(X'V^{-1}X)^- X'V^{-1}$. Similarly, the information matrix of REML is given by

$$\mathcal{I}_{REij}(\alpha, \beta) = -\mathbb{E}\Big[\frac{\partial^2 l_{RE}(\alpha, \beta)}{\partial \alpha_i \partial \alpha_j}\Big] = \frac{1}{2}\mathrm{tr}\Big(P\frac{\partial V}{\partial \alpha_i} P\frac{\partial V}{\partial \alpha_j}\Big).$$

[4] www.math.mcgill.ca/keith/surfstat

(3) Reestimation. The fixed effects and the overall variances are reestimated again using REML. If the procedure were iterated, it would converge to the REML estimates. However, it will take forever. So only perform this iteration once.

6.3.5 Functional Mixed Effect Models

It is possible to extend a linear mixed effect model to incorporate a more complex nonlinear growth pattern by taking functional covariates into the model. Goldsmith et al. (2011) modeled the clinical outcome Y_i of the ith subject as

$$Y_i = X\beta + \int_0^1 W_i(p) f(p)\, dp + \epsilon_i, \tag{6.12}$$

where $W_i(p)$ is the functional observation at position p and f is the smooth functional parameter that has to be estimated. β is the fixed effect shared by all subjects. Equation (6.12) is related to the standard impulse-response model, which has been often used in modeling fMRI responses. In the standard impulse-response model, the impulse function W_i is usually discrete and the response Y_i is continuous.

The fixed effect model (6.12) can be further generalized by incorporating the subject-specific random effect terms (Goldsmith et al., 2012). Let the jth measurement of the ith subject be Y_{ij}. We also have the corresponding functional observations $W_{ij}(p)$. Then Y_{ij} is modeled as

$$Y_{ij} = X_{ij}\beta + Z_{ij}\gamma_i + \int_0^1 W_{ij}(p) f(p)\, dp + \epsilon_i, \tag{6.13}$$

where X_i is the fixed effects and Z_i is the subject-specific random effects. $f(p)$ is the vector of functional effect that has to be estimated. The functional effect $f(p)$ are population-level parameters and do not vary across different subjects. Using the Karhunen–Loeve (KL) expansion, $W_{ij}(p)$ can be decomposed as

$$W_{ij}(p) = \sum_k c_{ijk}\psi_k(p), \tag{6.14}$$

where c_{ijk} are uncorrelated random variables and ψ_k are KL-basis (Fukunaga and Koontz, 1970). The main limitation of this model is that it is based on a single scalar outcome per subject.

Zipunnikov et al. (2011a,b) proposed a more general functional mixed effect model:

$$Y_{ij}(p) = \mu(p) + W_{ij}(p), \tag{6.15}$$

where $\mu(p)$ is the population-level fixed functional effect and $W_i(p)$ is the subject-specific random functional effect. W_{ij} is then decomposed using the similar KL-decomposition (6.14).

6.4 Simulating Dependent Images

Simulating dependent images at the voxel level is the basis of the simulating dependent networks. The method presented here can be directly applicable to simulating dependent connectivity matrices. We will use the mixed effect models as the baseline model for simulating dependent images.

6.4.1 Simulating Twin Images

The mixed effects models are implemented in R and MATLAB packages as well as SurfStat. SPM, FSL, and AFNI also have mixed effects model routines but are limited to fMRI multilevel analysis, so they are not easily modifiable to more general settings. In this section, we briefly explain how mixed effect models can be scripted in R.

We simulate total 260 numbers. We assume there were 40 MZ, 40 DZ different sex and 40 same-sex twin pairs and 20 singles. The twin membership is indexed by dummy variables `D_MZ`, `D_DZ`, `D_Single`, which takes value 1 (member) or 0 (no member). `D_Twin` is additional integer valued dummy variable indicating if subjects belong to the same twin. The same twins are assigned the same integer value. In R, the dummy variables are generated as follows:

```
D_MZ=c(rep(1,80),rep(0,80),rep(0,80),rep(0,20))
D_DZ=c(rep(0,80),rep(1,80),rep(0,80),rep(0,20))
D_Single=c(rep(0,80),rep(0,80),rep(0,80),rep(1,20))
D_Twin = c(kronecker(1:120,c(1,1)),121:140)
```

`D_Twin` is an array of 260 numbers 1 1 2 2 3 3 4 4 $\cdots$ 120 120 121 122 $\cdots$ 139 140 indicating twin-ness.

We simulate the ground truth based on the model

$$y = 4 + b_{Twin} + \epsilon, \tag{6.16}$$

where b_{Twin} is twin-level noise distributed as $b_{Twin} \sim N(0, \sigma^2)$ and individual-level noise $\epsilon \sim N(0, \sigma_\epsilon^2)$. We used parameters $\sigma = 0.5, \sigma_\epsilon = 1$. The ground truth is generated as follows.

```
b_twin=c()
for (i in 1:120)
{b=0.5*rnorm(1)
b=rep(b,2)
b_twin=c(b_twin,b)
}
b_twin = c(b_twin, rep(0,20))
y= 4 + b_twin + rnorm(260)
```

b is the twin-level noise, so the twin pairs have exactly the same amount of noise. y is the subject-level measurement with the fixed baseline 4 and possible twin variation b_twin.

Then we fit the following mixed effects model

$$Y = \beta_0 + \beta_2 D_{MZ} + \beta_3 D_{DZ} + \beta_4 D_{Single} + Z\gamma + \epsilon.$$

to the ground truth data simulated from (6.16) (Figure 6.3): the R function lme can be used to perform the linear mixed effects model fit. To apply the lme routine for fitting the proposed model, we need to generate a data frame with a specific formula that specifies cluster structure:

```
myframe = data.frame(y,D_MZ,D_DZ,D_Single,D_Twin)
myframe = groupedData(y ~ D_MZ+D_DZ+D_Single
          | D_Twin, data=myframe)
```

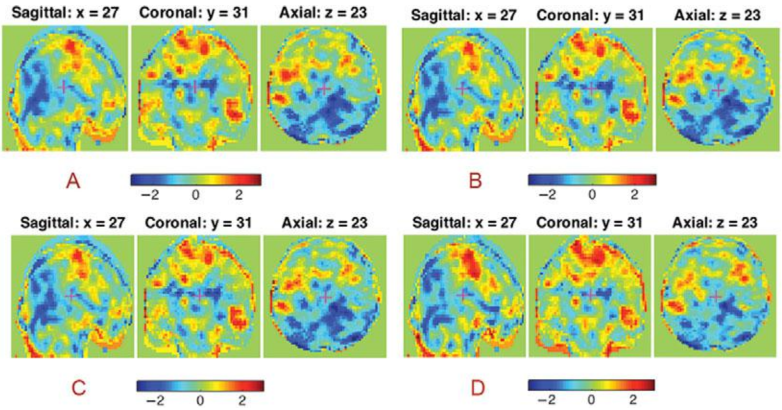


Figure 6.3 Comparison of fixed and mixed effects models on twin fMRI (Chung et al., 2017a). The figure displays the t-statistic maps of β_5 parameter in the models: (a) $y = \beta_0 + \beta_5 \cdot behavior$ (b) $y = \beta_0 + \beta_2 D_{MZ} + \beta_3 D_{DZ} + \beta_4 D_{Single} + \beta_5 \cdot behavior$ (c) $y = \beta_0 + \beta_5 \cdot behavior + b_{Twin}$ (d) $y = \beta_0 + \beta_2 D_{MZ} + \beta_3 D_{DZ} + \beta_4 D_{Single} + \beta_5 \cdot behavior + b_{Twin}$. The overall pattern of t-statistics maps is similar.

`myframe` indicates `y` variable depends on `D_MZ`, `D_DZ` and `D_Single` variables conditioning on `D_Twin`. Then the `lme` fit is done as follows:

```
results=lme(y~D_MZ+D_DZ+D_Single,data=myframe,
              random=~1|D_Twin)
summary(results)

Linear mixed effects model fit by REML
 Data: myframe
       AIC      BIC    logLik
  1269.591 1293.775 -628.7956

Random effects:
 Formula: ~1 | D_Twin
        (Intercept)  Residual
StdDev:   0.6249048 0.9096935

Fixed effects: y ~ 1 + D_MZ + D_DZ + D_Single
                Value Std.Error  t-value p-value
(Intercept)  4.071990 0.1268288 32.10619  0.0000
D_MZ        -0.080378 0.1553329 -0.51745  0.6054
D_DZ        -0.087891 0.1793630 -0.49002  0.6246
D_Single     0.206908 0.2774670  0.74570  0.4567
```

We obtained 4.07 as the parameter estimate for the intercept β_0. There is about 7% error in the estimation of the fixed effect term, which shows the mixed effects model is *not* necessarily a good model. The corresponding t-statistics and p-value are 32.1 and 0.0000. Since different statistical packages use different numerical schemes, it is expected to obtain different results if other packages are used.

6.4.2 Effect of Increased Sample Sizes

In this example, we increase the sample size substantially. We simulate 1,000 MZ, 500 DZ different sex and 500 same-sex twins and 200 singles. This gives the total sample size of 4,200. Also we reduce the size of signal to 0.1. This has the effect of reducing the t-statistic value. This simulates an extremely low signal-to-noise ratio setting. In R, the ground truth is generated as

```
y= 0.1 + b_twin + rnorm(4200)
```

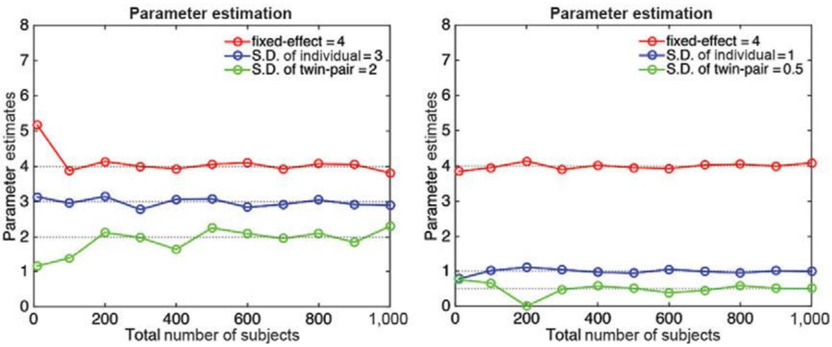


Figure 6.4 Performance of mixed effect model $y = \beta_0 + b_{Twin} + \epsilon$ with $b_{Twin} \sim N(0,\sigma^2)$ and $\epsilon \sim N(0,\sigma_\epsilon^2)$. Left: $\sigma = 2, \sigma_\epsilon = 3$. When the subject-level variability is larger than the twin-level variability, the mixed effect model does not perform well. Right: $\sigma = 0.5, \sigma_\epsilon = 1$.

The resulting R output after fitting `lme` is as follows:

```
> summary(results)
Linear mixed effects model fit by REML

Random effects:
 Formula: ~1 | D_Twin
        (Intercept)  Residual
StdDev:   0.4731649 0.9928018

Fixed effects: y ~ D_MZ + D_DZ + D_Single
              Value  Std.Error t-value p-value
(Intercept) 0.140478 0.037860   3.7104  0.0002
D_MZ       -0.026775 0.046369  -0.5774  0.5637
D_DZ       -0.045143 0.053542  -0.8431  0.3992
D_Single    0.031731 0.086493   0.3668  0.7138
```

The estimation is 0.14 for intercept β_0, which has 40% error (Figure 6.4). Thus, it is not likely that mixed effects models are reliable in a low signal-to-noise setting. As the signal-to-noise deceases more, it is expected the model fit performs worse.

6.5 Dependent Correlation Networks

In this section, we will show how to generate dependent correlation networks with complex dependent structures.

6.5.1 Simulating Correlation Networks

We use a linear regression model–based random network simulation approach (Chung et al., 2017a,d). We assume there are $p = 40$ nodes and $n = 5$ images or subjects. The data matrix $X_{n\times p} = (x_{ij})$ is simulated as standard normal in each component, i.e.,

$$x_{ij} \sim N(0,1). \tag{6.17}$$

It is more convenient to use vector and matrix forms in numerical implementation. Let

$$X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p).$$

Then (11.2.4) can be written as

$$\mathbf{x}_j \sim N(0, I).$$

We center and rescale the data vector $\mathbf{x}_j$ using `corr2norm.m`. Then the connectivity matrix is given by correlation matrix $X^\top X$. This is computed in MATLAB as follows

```
n=5; p=40;
X=normrnd(0, 1, n,p);
Xnorm = corr2norm(X1);
C1=Xnorm'*Xnorm;
```

Each time we repeat the preceding procedure, we are generating one instance of correlation network. Two independently generated correlation matrices are displayed in Figure 6.5. Since there is no dependency between nodes, these networks are considered as random networks with no signal.

6.5.2 Simulating a Single Module

The data matrix $X_{n\times p} = (x_{ij})$ is simulated as standard normal in each component, i.e.,

$$x_{ij} \sim N(0,1).$$

Let $Y = (y_{ij}) = (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_p) = X$. So far, there is no dependency (signal) between nodes in Y. We add dependency between nodes by letting

$$y_{ij} = \frac{1}{2} x_{i1} + N(0, 0.2^2)$$

for 10 nodes indexed by $j = 1, 2, \cdots, 10$. This can be equivalently written as

$$\mathbf{y}_j = \frac{1}{2}\mathbf{x}_1 + \cdot N(0, 0.2I).$$

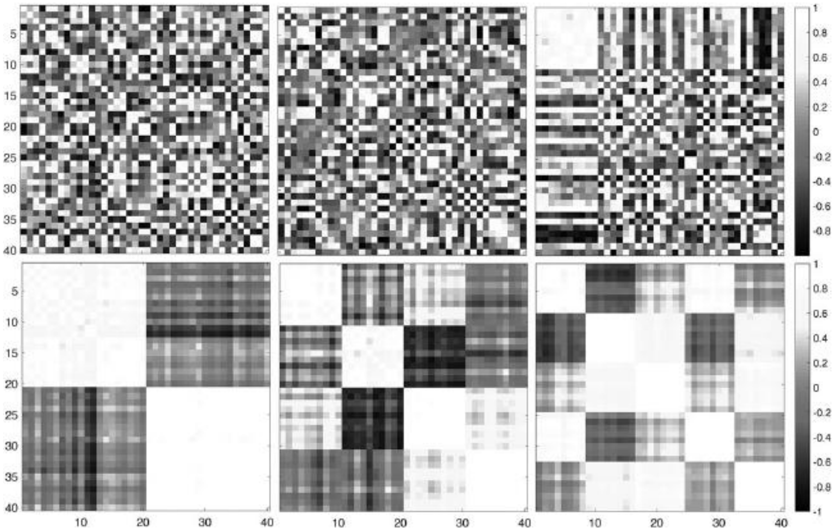


Figure 6.5 Top-left, middle: independently generated C_1. Top-right: C_2, Bottom: C_3 with two, four, and five modules.

This gives highly correlated 10 nodes in the upper-left block in the connectivity matrix (Figure 6.5). The procedure is implemented as

```
X=normrnd(0,1,n,p);
Y=X;
for i=1:10
    Y(:,i) =0.5*X(:,1)+normrnd(0,0.2,n,1);
end
Ynorm = corr2norm(Y);
C2=Ynorm'*Ynorm;
```

6.5.3 Simulating Multiple Modules

The details on simulating modular structure are given in (Chung et al., 2017d). The data matrix $X_{n\times p} = (x_{ij})$ is simulated as standard normal in each component, i.e.,

$$x_{ij} \sim N(0,1).$$

Let $Y = (y_{ij}) = X$. So far, there is no dependency (signal) between nodes in Y. Then we add the following block dependency:

$$\mathbf{y}_i = 0.5\mathbf{x}_{ci+1} + N(0, \sigma I_n).$$

This introduces modules in the network. We assumed there were total $k = 2, 4, 5$ modules and each module consists of $c = p/k$ number of points.

```
X=normrnd(0, 1, n,p);
Y=X;
k=4;
c=p/k;
for l=0:(k-1)
 for i=(1+l*c):(l+1)*c
   Y(:,i)=X(:,1+l*c)+normrnd(0,0.2,n,1);
 end
end
Ynorm = corr2norm(Y);
C3=Ynorm'*Ynorm;
```

Random networks with two, four, and five modules are displayed in Figure 6.5.

7
Persistent Homology

Persistent homology, a technique of computational topology (Carlsson and Memoli, 2008; Edelsbrunner and Harer, 2010), provides a coherent mathematical framework for quantifying brain networks. Instead of looking at networks at a fixed scale, persistent homology charts the changes in topological network features over multiple resolutions and scales (Zomorodian and Carlsson, 2005; Horak et al., 2009; Edelsbrunner and Harer, 2010). In doing so, it reveals the most *persistent* topological features, i.e., those that are robust to noise. This scale robustness is crucial since most brain network distances are parameter and scale dependent.

In persistent homology–based brain network analysis, instead of analyzing networks at one fixed threshold that may not be optimal, we build the collection of nested networks over every possible threshold using the *graph filtration*, a persistent homological construct (Lee et al., 2011a, 2012; Chung et al., 2013, 2015a). The graph filtration is a threshold-free framework for analyzing a family of graphs but requires hierarchically building specific nested subgraph structures. The graph filtration shares similarities to the existing multi-thresholding or multiresolution network models that use many different arbitrary thresholds or scales (Achard et al., 2006; He et al., 2008; Supekar et al., 2008; Lee et al., 2012; Kim et al., 2015). Such approaches are mainly used to visually display the dynamic pattern of how graph theoretic features change over different thresholds and the pattern of change is rarely quantified. Persistent homology can be used to quantify such dynamic patterns in a more coherent mathematical framework.

Persistent homology is an emerging technique and has been applied to brain network analysis only recently. There are very few validation and comparison studies against existing methods. However, in each of the limited comparison studies, the method is shown to very robust and outperforming many existing network measures and methods. In Lee et al. (2011a, 2012), persistent homology was shown to outperform against eight existing graph theory features such as assortativity, between centrality, clustering coefficient, characteristic path length, samll-worldness, modularity and global network homogeneity. In Chung et al. (2017d), persistent homology was shown to outperform L_1, L_2, and L_∞ matrix norms. In Wang et al. (2018), a persistent homology feature called persistent landscape was shown to outperform power spectral density and local variance methods. In Wang et al. (2017), persistent homology was shown to outperform topographic power maps. In Yoo et al. (2017), center persistency was shown to outperform the network-based statistic and elementwise multiple corrections.

7.1 Simplicial Homology

A high dimensional object can be approximated by the point cloud data X consisting of p number of points. If we connect points of which distance satisfy a given criterion, the connected points start to recover the topology of the object. Hence, we can represent the underlying topology as a collection of the subsets of X that consists of nodes that are connected (Hart, 1999; Edelsbrunner and Harer, 2010).

Definition 7.1 *Suppose $U \subset 2^X$ is the collection of all possible subsets of X. Then (X,U) is a topological space on X if*

1. $\emptyset, X \subset U$,
2. $u_1, u_2 \subset U$ implies $u_1 \cup u_2 \subset U$, and
3. $u_1 \cap u_2 \subset U$.

Note that every metric space is a topological space. In general, given a point cloud data set X with a rule for connections, the topological space is a simplicial complex and its element is a simplex (Zomorodian, 2009). For point cloud data, the Delaunay triangulation is probably the most widely used method for connecting points. The Delaunay triangulation represents the collection of points in space as a graph whose face consists of triangles. Another way of connecting point cloud data is based on the Rips complex often studied in persistent homology.

Homology is an algebraic formalism to associate a sequence of objects with a topological space (Edelsbrunner and Harer, 2010). In persistent homology, the algebraic formalism is usually built on top of objects that are hierarchically nested, such as morse filtration, graph filtration, and dendrograms. Formally, homology usually refers to homology groups that are often built on top of a simplicial complex for point cloud and network data (Lee et al., 2014).

Definition 7.2 *The k-simplex σ is the convex hull of $v+1$ independent points $v_0, \cdots, v_k$.*

A point is a 0-simplex, an edge is a 1-simplex, and a filled-in triangle is a 2-simplex. A simplicial complex is a collection of points (0-simplex), lines (1-simplex), triangles (2-simplex), and higher-dimensional counterparts. More formally,

Definition 7.3 *A* simplicial complex *K is a finite collection of simplices satisfying (Edelsbrunner and Harer, 2010) the following:*

1. *Any face of $\sigma \in K$ is also in K.*
2. *For $\sigma_1, \sigma_2 \in K$, $\sigma_1 \cap \sigma_2$ is a face of both σ_1 and σ_2.*

Hence a graph is a simplicial complex consisting of 0-simplices (nodes) and 1-simplices (edges). There are various simplicial complexes. One of them is the Rips complex.

Let C_k be the collection of k-simplices. Then we can define the kth boundary operator

$$\partial_k : C_k \to C_{k-1}$$

that removes the filled-in interior of k-simplices. Consider a filled-in triangle $\sigma = [v_1, v_2, v_3] \in C_2$ with three vertices v_1, v_2, v_3. Then, the boundary operator ∂_k applied to σ resulted in the collection of three edges that forms the boundary of σ:

$$\partial_2 \sigma = [v_1, v_2] + [v_2, v_3] + [v_3, v_1] \in C_1. \tag{7.1}$$

If we give the direction or orientation to edges such that

$$[v_3, v_1] = -[v_1, v_3],$$

we can write (7.1) as

$$\partial_2 \sigma = [v_1, v_2] + [v_2, v_3] - [v_1, v_3]. \tag{7.2}$$

If we use edge notation $e_{ij} = [v_i, v_j]$, (7.2) can be written in a more compact form:

$$\partial_2 \sigma = e_{12} + e_{23} - e_{13}. \tag{7.3}$$

We can apply the boundary operation ∂_1 further to $\partial_2\sigma$ and obtain

$$\begin{aligned}\partial_1\partial_2\sigma &= \partial_1 e_{12} + \partial_1 e_{23} - \partial_1 e_{13}\\ &= v_1 - v_2 + v_2 - v_3 - (v_1 - v_3) = 0.\end{aligned}$$

The boundary operation run twice will result in an empty set. Such algebraic representation for boundary operation has been very useful for effectively quantifying persistent homology.

7.1.1 Hodge Laplacian

The kth incidence matrix ∇_k is defined as the higher-dimensional version of incidence matrix in the graph theory (Lee et al., 2018a). Consider the boundary of $\sigma \in C_2$ consisting of edges. The incidence matrix ∇_2 will encode information on how edges are meeting to form the complex σ. The convention is that the columns of ∇_2 are the index for edges and the rows of ∇_2 are the index 2th simplicial complex. The (i, j)th entry of ∇_2 is 1 if the sign is edge is 1, -1 if the sign is -1, and 0 otherwise. Thus (7.2) is represented as

$$\nabla_2 = \begin{array}{c} e_{12} \\ e_{23} \\ e_{13} \end{array} \left(\begin{array}{c} 1 \\ 1 \\ -1 \end{array} \right),$$

where the column indicates the filled-in triangle $\sigma = [v_1, v_2, v_3]$.

Consider C_1 complex $\partial_2\sigma$ (7.3) that is an unfilled triangle. The incidence matrix ∇_1 of $\partial_2\sigma$ that encodes how nodes v_1, v_2, v_3 are forming edges e_{12}, e_{23}, e_{13} is given by

$$\nabla_1 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} e_{12} & e_{23} & e_{13} \end{array} \\ \left(\begin{array}{ccc} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & -1 & -1 \end{array} \right) \end{array}.$$

Similarly, the incidence matrix ∇_0 can be defined. Then the standard graph Laplacian is defined as

$$L_0 = \nabla_0 \nabla_0^\top,$$

which is also called the 0th Hodge Laplacian (Lee et al., 2018a). In general, the kth Hodge Laplacian is defined as

$$L_k = \nabla_{k+1}\nabla_{k+1}^\top + \nabla_k \nabla_k^\top.$$

The 0th Betti number, the number of connected components, is characterized by L_0 while the first Betti number, the number of cycles, is characterized by L_1.

7.1.2 Rips Filtrations

The Rips complex has been the main building block for persistent homology and defined on top of the point cloud data (Ghrist, 2008).

Definition 7.4 *The* Rips complex *is a graph constructed by connecting two data points if they are within specific distance* ϵ*.*

Figure 7.1 shows an example of the Rips complex that approximates the gray object with a point cloud. Given a point cloud data X, the Rips complex $R_X(\epsilon)$ is a simplicial complex whose k-simplices correspond to unordered $(k+1)$-tuples of points, which are pairwise within distance ϵ (Ghrist, 2008). While a graph has at most 1-simplices, the Rips complex has at most k-simplices. One major problem of the Rips complex is that given n points, it exactly produces a graph with n nodes so the resulting graph becomes very complicated when n becomes large.

The Rips complex has the property that

$$R_X(\epsilon_0) \subset R_X(\epsilon_1) \subset R_X(\epsilon_2) \subset \cdots$$

for $0 = \epsilon_0 \leq \epsilon_1 \leq \epsilon_2 \leq \cdots$. When $\epsilon = 0$, the Rips complex is simply the point cloud V. By increasing the ϵ-value, we are connecting more nodes so the size of the edge set increases. Such the nested sequence of the Rips

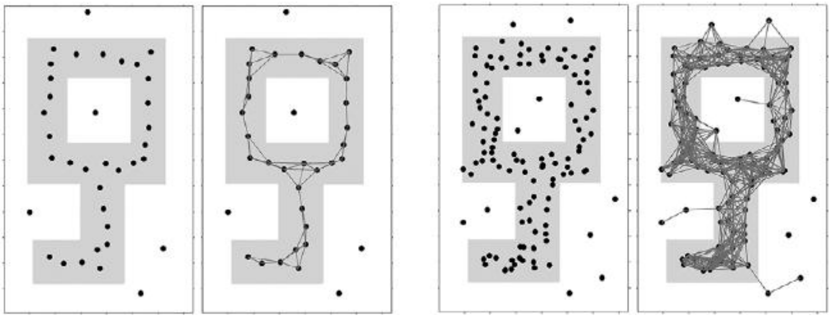


Figure 7.1 Left: a point cloud that approximates the underlying object (gray). Right: the Rips complex of the point cloud data. $\epsilon = 70$ is used for construction. If two points are within the ϵ radius, we connect them with a link. There are 36 nodes and 56 edges. Three outlying data are not connected to the largest connected component.

complexes is called a Rips filtration, the main object of interest in the persistent homology (Edelsbrunner and Harer, 2008). The increasing ϵ values are called the filtration values.

7.1.3 Betti Numbers

We build a vector space C_k using the set of k-simplices as a basis. The vector spaces $C_k, C_{k-1}, C_{k-2}, \cdots$ are then sequentially nested by boundary operator ∂_k (Edelsbrunner and Harer, 2010):

$$\cdots \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} C_{k-2} \xrightarrow{\partial_{k-2}} \cdots .$$

Let B_k be a collection of boundaries obtained as the image of ∂_k. Let Z_k be a collection of cycles obtained as the kernel of ∂_k. We have $B_k \subset Z_k$ and quotient space $H_k = Z_k/B_k$, which is called the kth homology group.

The elements of the kth homology group are often referred to as k-dimensional holes. The kth Betti number is then the number of k-dimensional holes. Equivalently, the rank of H_k is the kth Betti number. The Betti numbers are usually algebraically computed using Gaussian elimination. The zeroth Betti number is the number of connected components while the first Betti number is the number of cycles.

Persistent homology does not scale well with increased network size (Figure 7.2). The computational complexity of computing the kth Betti number depends on the number of k-simplices (Topaz et al., 2015). For n nodes, there are at most $O(n^{k+1})$ k-simplices. The homology computation requires Gaussian elimination, which often runs in $O(m^3)$, where m is the actual number of k-simplices. Thus, the overall computational complexity in the worst case is $O(n^{3k+3})$. As the number of nodes increases, the computational bottleneck can be severe (Figure 7.2).

Due to the computational bottleneck, resampling based statistical inference may not be feasible for extremely large-scale networks. There is a strong need to develop a faster inference procedure that does not rely on resampling techniques (Chung et al., 2017a).

Particularly for graphs and networks, β_1 can be computed easily as a function of β_0. Note that the Euler characteristic χ can be computed in two different ways

Theorem 7.1

$$\begin{aligned}\chi &= \beta_0 - \beta_1 + \beta_2 - \cdots \\ &= \#nodes - \#edges + \#faces - \cdots ,\end{aligned}$$

where $\#nodes, \#edges, \#faces$ are the number of nodes, edges, and faces.

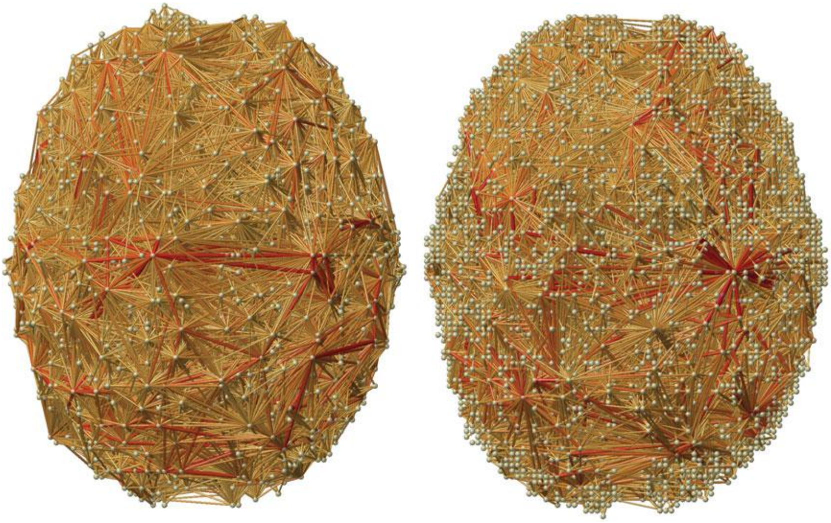


Figure 7.2 rs-fMRI correlation network of two subjects from HCP with more than 25,000 nodes. Identifying cycles and computing the number of cycles can be computationally demanding in this type of dense correlation network since persistent homology computations are not very scalable.

However, graphs do not have filled faces, Betti numbers higher than β_0 and β_1 and the number of elements beyond faces can be ignored. Thus, a graph with p nodes and l edges is given by

$$\chi = \beta_0 - \beta_1 = p - l.$$

Thus,

$$\beta_1 = p - l - \beta_0.$$

In graph filtration, we can show that β_0 and β_1 are monotonic.

Identifying connected components in a network is important to understand in decomposing the network into disjoint subnetworks. The number of connected components of a graph is a topological invariant that measures the number of structurally independent or disjoint subnetworks. The connected components can be identified using various techniques, including the Dulmage–Mendelsohn decomposition (Pothen and Fan, 1990), which has been widely used for decomposing sparse matrices into block triangular forms in speeding up matrix operations.

7.2 Morse Filtrations

A function is called a *Morse function* if all critical values are distinct and non-degenerate, i.e., the Hessian does not vanish (Milnor, 1973). For a 1D Morse function $y = f(t)$, define sublevel set $R(y)$ as

$$R(y) = f^{-1}(-\infty, y] = \{t \in \mathbb{R} : f(t) < y\}.$$

The sublevel set is the domain of f satisfying $f(t) \leq y$. As we increase y from $-\infty$, the sublevel set gets bigger such that

$$R(y_1) \subset R(y_2) \subset R(y_3) \subset \cdots$$

for

$$y_1 \leq y_2 \leq y_3 \leq \cdots .$$

The sequence of the sublevel sets forms a *Morse filtration*. The sequence of numbers $y_1, y_2, y_3, \cdots$ is called the *filtration values*.

Let $\beta_0(y)$ be the number of connected components of $R(y)$, which is a function of y. The number of connected components is usually called the zeroth Betti number and it is the most often used topological invariant in applications (Edelsbrunner and Harer, 2008). $\beta_0(y)$ only changes its value as it passes through critical values. The birth and death of connected components in the Morse filtration is characterized by the pairing of local minimums and maximums (Chung et al., 2009a).

Let us denote the local minimums as $g_1, \cdots, g_m$ and the local maximums as $h_1, \cdots, h_n$. Since the critical values of a Morse function are all distinct, we can combine all minimums and maximums and reorder them from the smallest to the largest: We further order all critical values together and let

$$g_1 = z_{(1)} < z_{(2)} < \cdots < z_{(m+n)} = h_n,$$

where z_i is either h_i or g_i and $z_{(i)}$ denotes the ith largest number in $z_1, \cdots, z_{m+n}$. It is left as an exercise to show that g_1 is smaller than h_1 and g_m is smaller than h_n in the unbounded domain $\mathbb{R}$.

By keeping track of the birth and death of components, it is possible to compute topological invariants of sublevel sets such as the zeroth Betti number β_0 (Edelsbrunner and Harer, 2008). As we move y from ∞ to ∞, at a local minimum, the sublevel set adds a new component so that

$$\beta_0(g_i - \epsilon) = \beta_0(g_i) + 1$$

for sufficiently small ϵ. This process is called the *birth* of the component. The newly born component is identified with the local minimum g_i.

Similarly for at a local maximum, two components are merged as one so that

$$\beta_0(h_i - \epsilon) = \beta_0(h_i) - 1.$$

This process is called the *death* of the component. The number of connected components will only change if we pass through critical points, and we can iteratively compute β_0 at each critical value as

$$\beta_0(z_{(i+1)}) = \beta_0(z_{(i)}) \pm 1.$$

The sign depends on if $z_{(i)}$ is maximum (-1) or minimum $(+1)$. This is the basis of the Morse theory (Milnor, 1973) that states that the topological characteristics of the sublevel set of the Morse function are completely characterized by critical values.

To reduce the effect of low signal-to-noise ratio and to obtain smooth Morse function, either spatial or temporal smoothing has been often applied to brain imaging data before persistent homology is applied. In (Chung et al., 2015a), and (Lee et al., 2017), Gaussian kernel smoothing was applied to 3D volumetric images. In (Wang et al., 2018), diffusion was applied to temporally smooth data.

7.2.1 Persistent Diagrams

The birth and death of connected components in the sublevel set of a function can be quantified and visualized by persistent diagrams (PD) (Figure 7.3). Consider a smooth Morse function with unique critical values A, B, C, D, E, F, and G. The dotted horizontal line is the threshold y that moves from $-\infty$ to ∞. Before the line hits the point A, the sublevel set of the function is empty except for the boundaries. Each time the line touches the local minimum A, B, and D, a new component that contains the local minimum is born. Each time the line touches the local maximum C, D, and E, the two components merge together. This is considered as the death of a component.

Following the *Elder Rule*, when we pass a maximum and merge two components, we pair the maximum (birth) with the higher of the minimums of the two components (birth) (Zomorodian and Carlsson, 2005; Edelsbrunner and Harer, 2008, 2010). Doing so, we are pairing the birth of a component to its death. However, when we include the boundaries of the domain, we also need to take care of the birth and death of the boundaries and accordingly pair them.

In Figure 7.3, we pair point C (death) with point B (birth). Once we move up to the next maximum E, we pair point E (death) with A (birth). When we move up maximum F, we pair F (death) with minimum D (birth). At the last

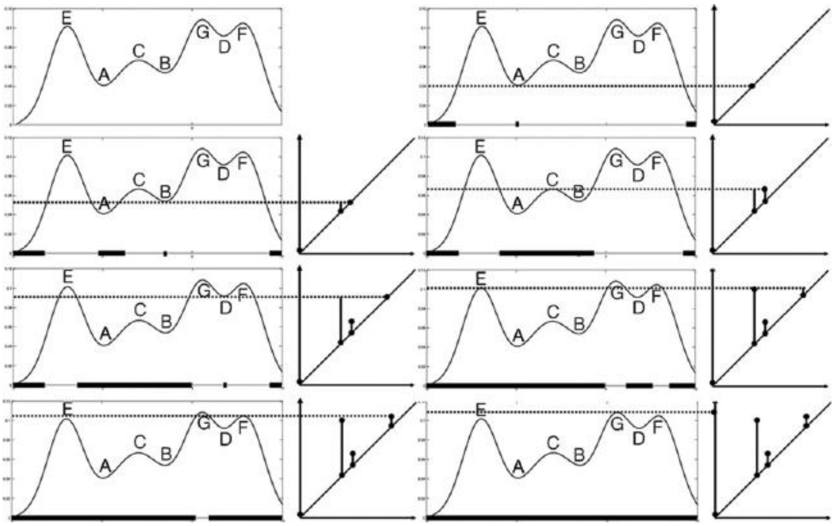


Figure 7.3 The Morse filtration and corresponding persistent diagram (PD) encoding the birth and death times of the connected components over the filtration. Yuan Wang of University of Wisconsin–Madison generated the figure (Wang et al., 2018).

maximum G, we pair it with the left boundary, which is set at 0 in this example. The line segment that shows the time of birth and death in the y-axis is called the *barcodes*. The scatter points of the time of birth in x-axis and the time of death in the y-axis is called in the *persistence diagram* (see Figure 7.3).

For higher-dimensional Morse functions, saddle points can also create or merge sublevel sets, so we also have to be concerned with them. Since there are no saddle points in 1D Morse functions, we don't need to worry about saddle points in 1D functional signals. The addition of the saddle points makes the construction of the persistence diagrams much more complex. However, the saddle points do not yield a clear statistical interpretation compared to local minimums and maximums. In fact, there are no statistical methods or applications developed for saddle points in the literature.

Given functional signal f observed at time points $t_1, \cdots t_n$, the critical points can be numerically estimated and obtained by checking sign changes in the differences

$$f(t_{i+1}) - f(t_i).$$

The persistent diagrams and equivalent barcodes are the original most often used descriptors in persistent homology. The persistent diagrams possess desirable bounding properties such as Lipschitz stability with respect to the bottleneck distance. Unfortunately, statistical analysis on persistent diagrams are difficult since it is not a clear-cut to define the average persistent diagram. Even the Fréchet mean is not unique, rendering it a challenging statistical issue to perform inference on directly on persistent diagrams (Chung et al., 2009a; Heo et al., 2012). To remedy the problem, persistent landscape was proposed in (Bubenik, 2015).

Given a barcode represented as an interval $I = [a,b]$, we can define the piecewise linear bump function $h_I : \mathbb{R} \to \mathbb{R}$ by

$$h_I(\lambda) = \max(\min(\lambda - a, b - \lambda), 0).$$

The geometric representation of the bump function (7.4) is a right-angled isosceles triangle with height equal to half of the base of the corresponding interval in the barcode. With this new representation, it is possible to average persistent landscapes.

7.2.2 Inference on Persistent Diagrams

Stability of persistent diagram. The stability of the persistent diagram under small perturbation is established in Cohen-Steiner et al. (2007) and Edelsbrunner and Harer (2008). Let $D(\mu)$ and $D(\nu)$ be the persistence diagrams of μ and ν respectively. A metric on the space of persistence diagrams is the bottleneck distance d that bounds the Hausdorff distance. It is given by

$$d(D(\mu), D(\nu)) = \inf_{\gamma} \sup_{p \in D(\mu)} \|p - \gamma(p)\|_\infty, \tag{7.4}$$

where the infimum is taken over all bijections $\gamma : D(\mu) \to D(\nu)$ and $\| \cdot \|_\infty$ is the sup-norm metric. In Cohen-Steiner et al. (2007), the following result is proven:

$$d(D(\mu), D(\nu)) \le \|\mu - \nu\|_\infty. \tag{7.5}$$

Inference on persistent diagrams. Given a persistent diagram in the square $[0, L]^2$, we can discretize the square with the uniform grid (Chung et al., 2009a). A concentration map is obtained by counting the number of points in each pixel. Notice that this approach is somewhat similar to the voxel-based morphometry (Ashburner and Friston, 2000), where brain tissue density maps are used as a shapeless metric for characterizing concentration of the amount of tissue. The inference corrected for multiple comparisons is then done by performing the permutation test on the t-statistic of concentration maps. If data are white noise, points in PD should occur close to the diagonal line, i.e., $y = x$. The deviation from $y = x$ indicates signal.

Likely, the concentration maps do not follow Gaussian distributional assumptions. Note that this is the usual multiple comparison problem (Worsley et al., 1996b) due to correlated t-statistic values across neighboring pixels. We can perform the permutation test to empirically estimate the distribution of

$$\sup_{t \in [0,L]^2} T(t) \text{ and } \inf_{t \in [0,L]^2} T(t)$$

to determine the statistical significance. By thresholding the tails of the empirical distribution at significance 0.05, we can obtain the corresponding quantile points.

7.2.3 Why Critical Values?

Persistent diagrams are the plots of pairing of critical values of a function. The use of critical values of measurements within classical image analysis and computer vision has been relatively limited so far, and typically appears as part of simple preprocessing tasks such as feature extraction and identification of edge pixels in an image. For example, first- or second-order image derivatives may be used to identify the edges of objects to serve as the contour of an anatomical shape, possibly using priors to provide additional shape context. Specific properties of critical values as a topic on its own, however, have received less attention. Whether critical points may serve a more central role in the design of image processing algorithms is a question that has not been investigated in sufficient detail.

Critical points of measurements are not used in classical image analysis and computer vision very frequently. One reason is that it is difficult to construct a streamlined linear analysis framework using critical points, or values of images. Also, the computation of critical values is a nonlinear process and almost always requires the numerical estimation of derivatives. In some applications where this is necessary, the discretization scheme must be chosen carefully, and remains an active area of research (Osher and Fedkiw, 2003). It is noticed that in most of these applications, the interest is only in the stable estimation of these points rather than their properties, and how these properties vary as a function of images.

In brain imaging, on the other hand, the use of extreme values has been quite popular in other types of problems. For example, these ideas are employed in the context of multiple comparison correction using random field theory (Worsley et al., 1996b; Kiebel et al., 1999; Taylor and Worsley, 2008). Recall that in the random field theory, the extreme value of a statistic is obtained from an ensemble of images and is used to compute the p-value for correcting for correlated noise across neighboring voxels.

Critical points have been also been used in image processing, and serve as tools for feature extraction (Cootes et al., 1993; Sato et al., 1998; Antoine et al., n.d.). In this context, image derivatives are computed after image smoothing and thresholded to obtain edges and ridges of images that are used to identify pixels likely to lie on boundaries of anatomical objects. Then the collection of critical points is used as a geometric feature that characterizes anatomical shape.

7.3 Graph Filtrations

In persistent homological brain network analysis as first established in Lee et al. (2011a, 2012), instead of analyzing networks at one fixed threshold, we build the collection of nested networks over every possible threshold using a *graph filtration*, a persistent homological construct (Lee et al., 2012; Chung et al., 2013). The graph filtration is a threshold-free framework for analyzing a family of graphs but requires hierarchically building nested subgraph structures. The graph filtration framework shares similarities to existing multithresholding or multiresolution network models that use many different arbitrary thresholds or scales (Achard et al., 2006; He et al., 2008; Lee et al., 2012; Kim et al., 2015). However such approaches are exploratory, being mainly used to visualize graph feature changes over different thresholds without quantification. Persistent homology, on the other hand, quantifies such feature changes in a coherent way.

Euclidean distance is an often used metric in building filtrations in persistent homology. For point cloud data, the Euclidean distance between points is used to build the Rips filtration (Edelsbrunner and Harer, 2010). Numerous brain network studies used the Euclidean distances for building network filtrations (Lee et al., 2011b, 2012; Petri et al., 2014; Khalid et al., 2014; Cassidy et al., 2015; Chung et al., 2015a, 2017a; Anirudh et al., 2016; Wong et al., 2016; Palande et al., 2017).

7.3.1 Filtration on Weighted Graphs

Recently a concept of graph filtration has been proposed and successfully applied to brain networks as a way to quantify networks without thresholding (Lee et al., 2012; Chung et al., 2017a).

Definition 7.5 *Given weighted network $\mathcal{X} = (V, w)$ with edge weight $w = (w_{ij})$, the binary network $\mathcal{X}_\epsilon = (V, w_\epsilon)$ is a graph consisting of the node set V and the binary edge weights w_ϵ given by*

$$w_\epsilon = (w_{\epsilon,ij}) = \begin{cases} 1 & \textit{if } w_{ij} > \epsilon; \\ 0 & \textit{otherwise.} \end{cases} \tag{7.6}$$

Note Lee et al. (2012) define the binary graphs by thresholding above, which is consistent with the definition of the Rips filtration. However, in brain imaging, higher value w_{ij} indicates stronger connectivity. Thus, we are thresholding below (Chung et al., 2015a).

Note w_ϵ is the adjacency matrix of $\mathcal{X}_\epsilon$, which is a simplicial complex consisting of 0-simplices (nodes) and 1-simplices (edges) (see Figure 7.4) (Ghrist, 2008). In the metric space $\mathcal{X} = (V, w)$, the Rips complex $\mathcal{R}_\epsilon(X)$ is a simplicial complex whose $(p-1)$-simplices correspond to unordered p-tuples of points that satisfy $w_{ij} \leq \epsilon$ in a pairwise fashion (Ghrist, 2008). While the binary network $\mathcal{X}_\epsilon$ has at most 1-simplices, the Rips complex can have at most $(p-1)$-simplices (Figure 7.4). Thus, the complement of the binary graph $\mathcal{X}_\epsilon^c \subset \mathcal{R}_\epsilon(\mathcal{X})$. The Rips complex has the property that

$$\mathcal{R}_{\epsilon_0}(\mathcal{X}) \subset \mathcal{R}_{\epsilon_1}(\mathcal{X}) \subset \mathcal{R}_{\epsilon_2}(\mathcal{X}) \subset \cdots$$

for $0 = \epsilon_0 \leq \epsilon_1 \leq \epsilon_2 \leq \cdots$. When $\epsilon = 0$, the Rips complex is simply the node set V. By increasing the filtration value ϵ, we are connecting more nodes so the size of the edge set increases. Such a nested sequence of the Rips complexes is called a Rips filtration, the main object of interest in the persistent homology (Edelsbrunner and Harer, 2008).

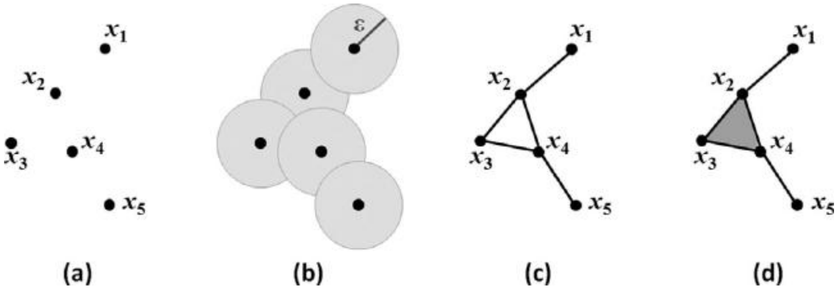


Figure 7.4 The difference between a binary network and a Rips complex. (a) Point cloud data X. (b) The ball of radius ϵ centered at each point. (c) Binary network $X|_\epsilon$. (d) Rips complex $\mathcal{R}_\epsilon(X)$. Unlike the binary network, the Rips complex has a filled-in triangle. The figure was generated by Hyekyoung Lee of Seoul National University.

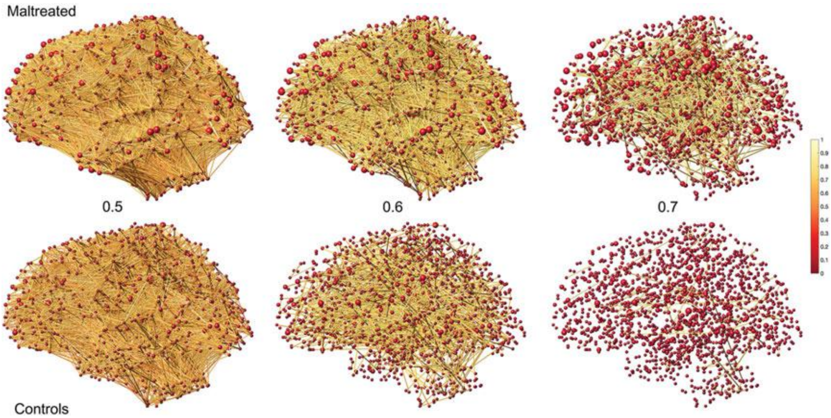


Figure 7.5 Graph filtrations of maltreated children vs. normal control subjects on FA-values (Chung et al., 2015a). The Pearson correlation is used as filtration values at 0.5, 0.6, and 0.7. Maltreated subjects show much higher correlation of FA-values, indicating a more homogeneous and less varied structural covariate relationship.

Since a binary network is a special case of the Rips complex, we also have

$$\mathcal{X}_{\epsilon_0}^c \subset \mathcal{X}_{\epsilon_1}^c \subset \mathcal{X}_{\epsilon_2}^c \subset \cdots$$

for $0 = \epsilon_0 \leq \epsilon_1 \leq \epsilon_2 \cdots$. Equivalently, we also have

$$\mathcal{X}_{\epsilon_0} \supset \mathcal{X}_{\epsilon_1} \supset \mathcal{X}_{\epsilon_2} \supset \cdots .$$

The sequence of such nested multiscale graphs is defined as the *graph filtration* (Lee et al., 2011a, 2012). Figure 7.5 shows an example of graph filtration in a fluorodeoxyglucose (FDG) positron emission tomography (PET) study of three different populations (Lee et al., 2012).

Note that $\mathcal{X}_0$ is the complete weighted graph while $\mathcal{X}_\infty$ is the node set V. By increasing the threshold value, we are thresholding at higher connectivity so more edges are removed. Given a weighted graph, there are infinitely many different filtrations. This makes the comparisons between two different graph filtrations difficult. For network $\mathcal{Y} = (V, z)$ with the same node set but with different edge weight z, with different filtration values, $\lambda_0 \leq \lambda_1 \leq \lambda_2 \cdots$, we have

$$\mathcal{Y}_{\lambda_0} \supset \mathcal{Y}_{\lambda_1} \supset \mathcal{Y}_{\lambda_2} \supset \cdots .$$

Then we compare two different graph filtrations $\{\mathcal{X}_{\epsilon_i}\}$ and $\{\mathcal{Y}_{\lambda_i}\}$. For different ϵ_j and ϵ_{j+1}, we can have identical binary graph, i.e., $\mathcal{X}_{\epsilon_j} = \mathcal{X}_{\epsilon_{j+1}}$. So it is possible there is a unique filtration that can be used for comparisons. Let the *level of a filtration* be the number of nested unique sublevel sets in the given filtration (Chung et al., 2015a).

Theorem 7.2 *For graph $X = (V, w)$ with q unique positive edge weights, the maximum level of a filtration is $q+1$. Further, the filtration with $q+1$ filtration level is unique.*

Proof. For a graph with p nodes, the maximum number of edges is $(p^2 - p)/2$, which is obtained in a complete graph. If we order the edge weights in the increasing order, we have the sorted edge weights:

$$0 = w_{(0)} < \min_{j,k} w_{jk} = w_{(1)} < w_{(2)} < \cdots < w_{(q)} = \max_{j,k} w_{jk},$$

where $q \leq (p^2 - p)/2$. The subscript $_{(\,)}$ denotes the order statistic. For all $\lambda < w_{(1)}$, $\mathcal{X}_\lambda = \mathcal{X}_0$ is the complete graph of V. For all $w_{(r)} \leq \lambda < w_{(r+1)}$ $(r = 1, \cdots, q-1)$, $\mathcal{X}_\lambda = \mathcal{X}_{w_{(r)}}$. For all $w_{(q)} \leq \lambda$, $\mathcal{X}_\lambda = \mathcal{X}_{\rho_{(q)}} = V$, the vertex set. Hence, the filtration given by

$$\mathcal{X}_0 \supset \mathcal{X}_{w_{(1)}} \supset \mathcal{X}_{w_{(2)}} \supset \cdots \supset \mathcal{X}_{w_{(q)}}$$

is *maximal* in a sense that we cannot have any additional level of filtration. □

Throughout the book, the *maximal graph filtration* (7.7) will be mainly used. The condition of having unique edge weights in Theorem 7.2 is not restrictive in practice. Assuming edge weights to follow some continuous distribution, the probability of any two edges being equal is zero. For discrete distribution, it may be possible to have identical edge weights. Then simply add Gaussian noise or add extremely small increasing numbers

$$0, \frac{1}{10^{32}}, \frac{2}{10^{32}}, \cdots \frac{q}{10^{32}}$$

to q number of edges. Among many possible filtrations, we will use the maximal filtration (7.7) in the study since it is uniquely given. The finiteness and uniqueness of the filtration levels over finite graphs are intuitively clear by themselves and are implicitly assumed in software packages such as javaPlex (Adams et al., 2014). However, we still need a rigorous statement to specify the type of filtration we are using.

7.3.2 Node-Based Filtration

Instead of doing graph filtration at the edge level, it is possible to build graph filtration at the node level (Wang et al., 2017). Consider graph $G = (V, E)$

with nodes $V = 1, 2, \cdots, p$ and node weights w_i defined at each node i. With threshold λ, define a binary network $G_\lambda = (V_\lambda, E_\lambda)$, where

$$V_\lambda = \{i \in V : w_i \leq \lambda\}$$

and

$$E_\lambda = \{(i, j) \in E : \max(w_i, w_j) \leq \lambda\}.$$

Note $E_\lambda \subset E$ such that two nodes i and j are connected if $\max(w_i, w_j) \leq \lambda$. We include a node from G in G_λ when the threshold λ is above its weight, and we connect two nodes in G_λ with an edge when λ is above the larger weight of any of the two nodes. Suppose we have edge weights

$$w_{(1)} \leq w_{(2)} \leq \cdots \leq w_{(q)}$$

of G. Then we have the *node-based graph filtration*

$$G_{w_{(1)}} \subset G_{w_{(2)}} \subset \cdots \subset G_{w_{(q)}}. \tag{7.7}$$

Unlike Rips filtration, the filtration (7.7) is not affected by reindexing nodes since the edge weights remain the same regardless of node indexing. Each $G_{w_{(j)}}$ in (7.7) consists of clusters of connected nodes; as λ increases, clusters appear and later merge with existing clusters. The pattern of changing clusters in (7.7) has the following properties.

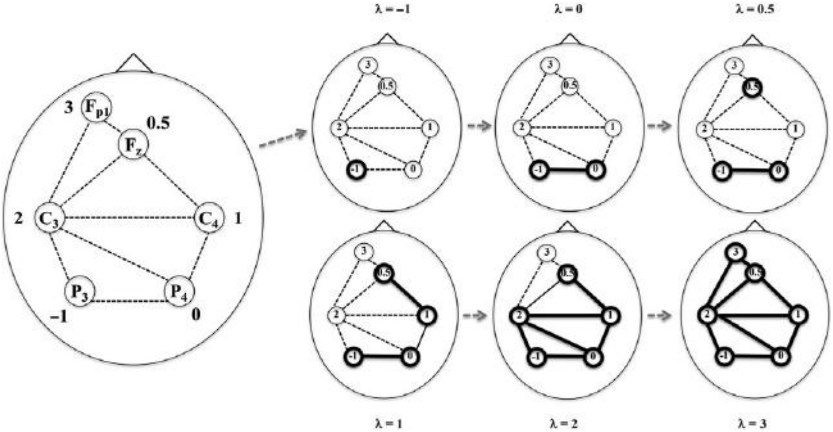


Figure 7.6 Schematic of the node-based filtration on six EEG channels in the international 10–20 system. (Left: the six-channel layout with the corresponding Delaunay triangulation on the powers at the EEG channels. Right: as the filtration value λ increases, we include the nodes and edges with weights less than λ. As λ increases, more nodes and edges are merged in the filtration. The figure was generated by Yuan Wang of University of South Carolina (Wang et al., 2017).

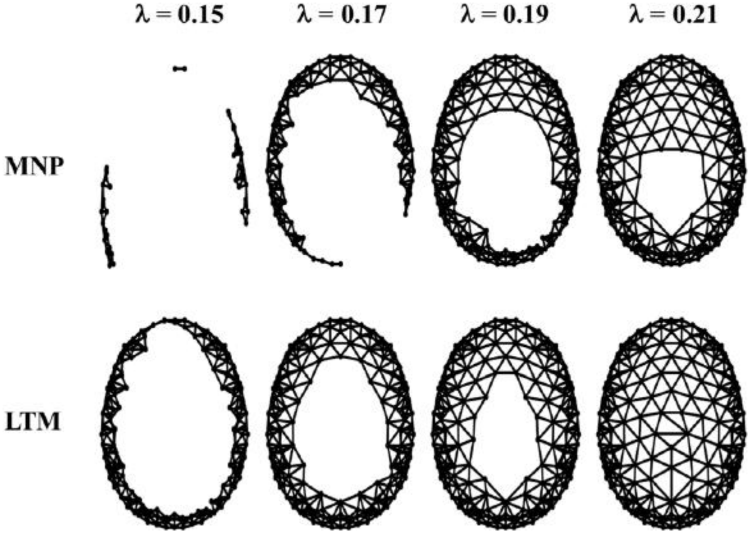


Figure 7.7 Filtrations on EEG power maps in the beta band during sleep in meditation native participants (MNP) and long-term mediators (LTM) showing significant filtration differences (Wang et al., 2017). The figure was generated by Yuan Wang of University of South Carolina.

For $w_{(i)} \leq \lambda < w_{(i+1)}$, $G_\lambda = G_{w_{(i)}}$, the filtration (7.7) is maximal in the sense that no more G_λ can be added to (7.7) (Chung et al., 2015a). As λ increases from $w_{(i)}$ to $w_{(i+1)}$, only the node v'_{i+1} that corresponds to the weight $w_{(i+1)}$ is added in $V_{w_{(i+1)}}$.

The node-level filtration is illustrated on a six-channel EEG layout in the international 10–20 system (Figure 7.6). We first build up the Delaunay triangulation over the six-channel layout. Node weights are the powers at the EEG channels. At each filtration value λ, we include the nodes and edges with weights less than or equal to λ. The clusters change as λ increases. Figure 7.7 shows the filtration difference in EEG power maps between meditation native participants (MNP) and long-term mediators (LTM) (Wang et al., 2017).

7.4 Betti Plots

The graph filtration can be quantified using monotonic function f satisfying

$$f(\mathcal{X}_{\epsilon_0}) \geq f(\mathcal{X}_{\epsilon_1}) \geq f(\mathcal{X}_{\epsilon_2}) \geq \cdots \tag{7.8}$$

or

$$f(\mathcal{X}_{\epsilon_0}) \leq f(\mathcal{X}_{\epsilon_1}) \leq f(\mathcal{X}_{\epsilon_2}) \leq \cdots . \tag{7.9}$$

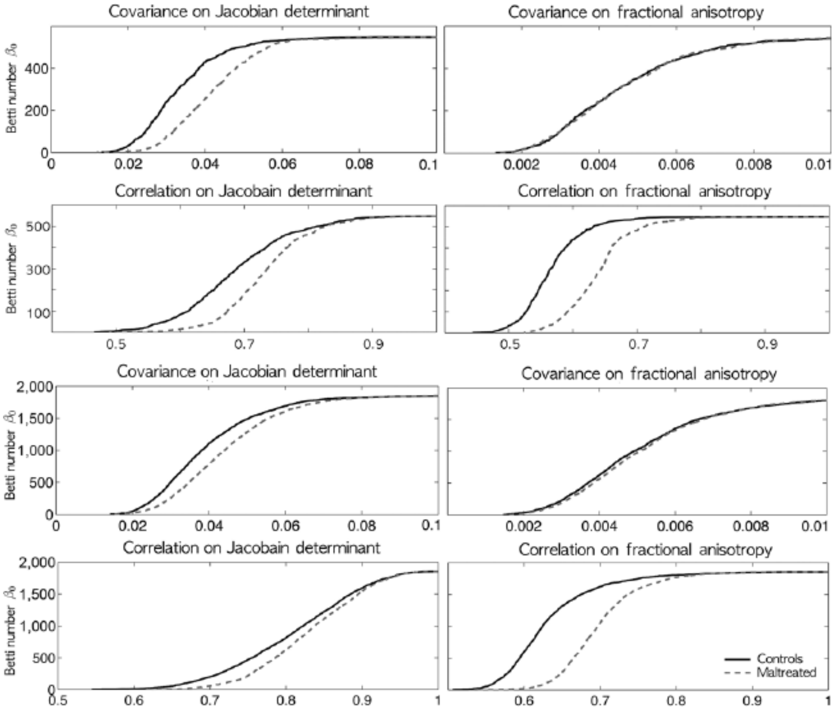


Figure 7.8 The Betti plots on the covariance correlation matrices for Jacobian determinant (left column) and fractional anisotrophy (right column) on 548 (top two rows) and 1,856 (bottom two rows) nodes (Chung et al., 2015a). Unlike the covariance, the correlation seems to shows huge group separation between normal and maltreated children visually. However, in all seven cases except the top-right (548 nodes covariance for FA), statistically significant differences were detected using the rank-sum test on the areas under the Betti plots (p-value < 0.001). The shapes of Betti plots are consistent between the studies with different node sizes, indicating the robustness of the proposed method over changing number of nodes.

The number of connected components (zeroth Betti number β_0) and the number of cycles (first Betti number β_1) satisfy the monotonicity (Figure 7.8). The size of the largest cluster (denoted as γ) satisfies a similar but opposite relation of monotonic increase. There are numerous monotone graph theory features (Chung et al., 2015a, 2017a).

Theorem 7.3 *In a graph, Betti numbers β_0 and β_1 are monotone over filtration on edge weights.*

Proof. When we do filtration on the maximal filtration in (7.7), edges are deleted one at a time. Since an edge has only two end points, the deletion of an edge disconnects the graph into at most two. Thus, the number of connected components (β_0) always increases, and the increase is at most by one. The Euler characteristic χ of the graph is given by (Adler et al., 2010).

$$\chi = \beta_0 - \beta_1 = p - q,$$

where p and q are the number of nodes and edges respectively. Thus,

$$\beta_1 = \beta_0 - p + q.$$

Note p is fixed over the filtration but q is decreasing by one while β_0 increases at most by one. Hence, β_1 always decreases, and the decrease is at most by one. □

In graph filtrations, the number of connected components increases as the filtration value increases. The pattern of increasing number of connected components can visually show how the topology of the graph changes over different parameter values. The overall pattern of *Betti (number) plots* can be used as a summary measure of quantifying how the graph changes over increasing edge weights (Chung et al., 2013) (Figure 7.9). The Betti number plots are related but different from barcodes in literature. The Betti number is equal to the number of bars in the barcodes at the specific filtration value. To construct Betti plots, it is not necessary to perform filtrations for infinitely

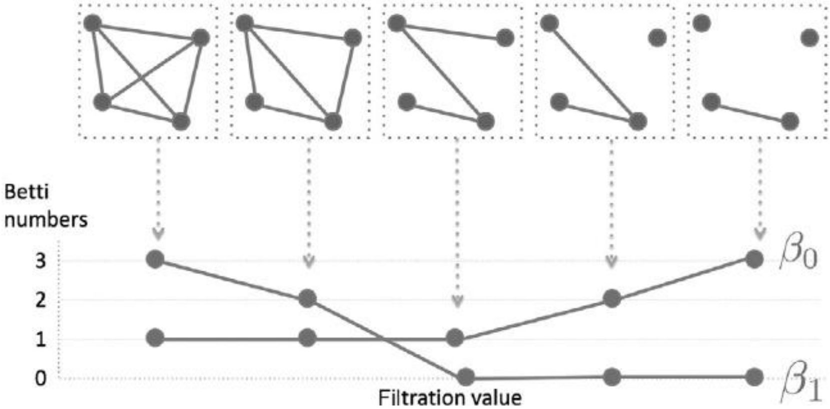


Figure 7.9 Schematic of graph filtration and Betti plots. We sort the edge weights in an increasing order. We threshold the graph at filtration values and obtain binary graphs. The thresholding is performed sequentially by increasing the filtration values. The zeroth Betti number β_0, which counts the number of connected components, and the first Betti number β_1, which counts the number of cycles, is then plotted over the filtration. The Betti plots are monotonic.

many possible λ values. From Theorem 7.2, the maximum possible number of filtration level for plotting the Betti numbers is one plus the number of unique edge weights. For a tree, which is a graph with no cycle, we can come up with a much stronger statement.

Theorem 7.4 *For a tree T with $p \geq 2$ nodes and unique positive edge weights $w_{(1)} < w_{(2)} < \cdots < w_{(p-1)}$, the plot for the first Betti number (β_0) corresponding to the maximal graph filtration is given by the coordinates*

$$(0,1),(w_{(1)},2),\cdots,(w_{(2)},3),(w_{(p-1)},p),(\infty,p).$$

Proof. For a tree T with p nodes, there are total $p-1$ edges. Then from Theorem 7.2, we have the maximal filtration

$$T_{w_{(0)}} \supset T_{w_{(1)}} \supset T_{w_{(2)}} \supset \cdots \supset T_{w_{(p-1)}}. \tag{7.10}$$

Since all the edge weights are above filtration value $w_{(0)} = 0$, all the nodes are connected, i.e., $\beta_0(w_{(0)}) = 1$. Since no edge weight is above the threshold $w_{(q-1)}$, $\beta_0(w_{(p-1)}) = p$. At each time we threshold an edge, the number of components increases exactly by one in the tree. Thus, we have

$$\beta_0(w_{(1)}) = 2, \beta_0(w_{(2)}) = 3, \cdots, \beta_0(w_{(p-1)}) = p.$$

□

For a general graph, it is not possible to analytically determine the coordinates for its Betti plot (see Figure 7.10). The best we can do is to compute the number of connected components β_0 numerically using the

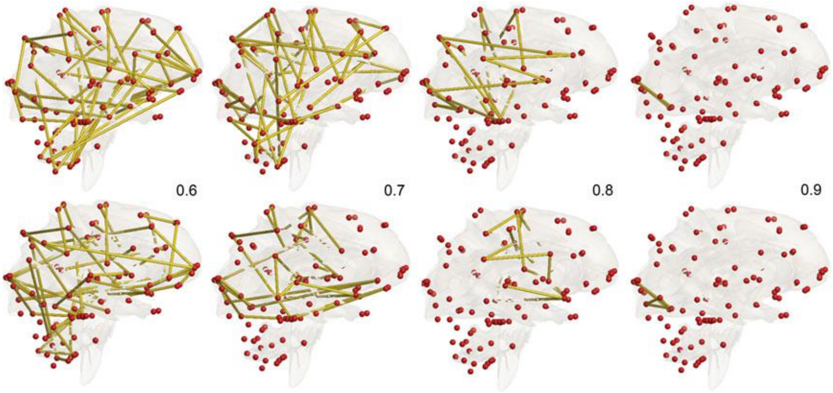


Figure 7.10 The largest cycle at given correlation thresholds on rs-fMRI. Two representative subjects in HCP were used (Gritsenko et al., 2018a). As the threshold increases, the length of cycles decreases monotonically.

single linkage dendrogram (SLD) method (Lee et al., 2012), the Dulmage–Mendelsohn decomposition (Pothen and Fan, 1990; Chung et al., 2011c) or existing simplicial complex approaches (Edelsbrunner et al., 2002; de Silva and Ghrist, 2007; Carlsson and Memoli, 2008).

7.4.1 Statistical Inference on Betti Number Plots

The zeros and the first Betti numbers can be used as features for characterizing network differences statistically. Suppose there are n subjects and p nodes in Group 1. For subject i, we have measurement x_{ij} at node j. Denote data matrix as $X = (x_{ij})$, where x_{ij} is the measurement for subject i at node j.

We then construct a connectivity matrix corresponding to X and the corresponding kth Betti number plots $\beta_k^1(\lambda)$ using X. Thus, $\beta_k^1(\lambda)$ is a function of X. Consider another Group 2 consists of m subjects. For Group 2, the data matrix is denoted as $Y = (y_{ij})$, where y_{ij} is the measurement for subject i at node j. Group 2 will also generate single Betti number plot $\beta_k^2(\lambda)$ as a function of Y. We are then interested in testing if the shapes of Betti number plots are different between the groups. The inference can be done by comparing the areas under the Betti plots. So the null hypothesis of interest is

$$H_0 : \int_0^1 \beta_k^1(\lambda)\, d\lambda = \int_0^1 \beta_k^2(\lambda)\, d\lambda \tag{7.11}$$

while the alternate hypothesis is

$$H_1 : \int_0^1 \beta_k^1(\lambda)\, d\lambda \neq \int_0^1 \beta_k^2(\lambda)\, d\lambda.$$

This inference avoids the use of multiple comparisons over the range of λ values. The null hypothesis (7.11) is related to the following pointwise null hypothesis:

$$H_0' : \beta_0^1(\lambda) = \beta_0^2(\lambda) \text{ for all } \lambda \in [0, 1]. \tag{7.12}$$

If the hypothesis (7.12) is true, the hypothesis (7.11) is also true; however, the inverse may not be true. Thus, testing the area under the curve is related to testing the height of the curve at every point. The advantage of using the area under the curve is that we do not need to worry about multiple comparisons associated with testing (7.12). The area under the curve seems a reasonable approach to use for Betti plots. A similar approach has been introduced in (Chung, 2001) in removing the multiple comparisons and producing a single summary test statistic.

There is no prior study on the statistical distribution on the Betti numbers so it is difficult to construct a parametric test procedure. Further, since there is only one Betti plot per group, it is necessary to empirically construct the null distribution and determine the p-value by resampling techniques such as the permutation test and jackknife (Efron, 1982; Lee et al., 2012; Chung et al., 2013). For this section, we use the jackknife resampling.

For Group 1 with n subjects, one subject is removed at a time and the remaining $n - 1$ subjects are used in constructing a network and a Betti plot. Let X_{-l} be the data matrix, where the lth row (subject) is removed from X. Then for each lth subject removed, we compute $\beta_k^{1(-l)}$, which is a function of λ and X_{-l}. Repeating this process for each subject, we obtain n Betti plots $\beta_k^{1(-1)}, \beta_k^{1(-2)}, \cdots, \beta_k^{1(-n)}$. For Group 2, the lth row (subject) is removed from the original data matrix Y and we obtain the data matrix as Y_{-l}. For each lth subject removed, we compute $\beta_k^{2(-l)}$, which is a function of λ and X_{-l}. Repeating this process for each subject, we obtain m Betti plots $\beta_k^{2(-1)}, \beta_k^{2(-2)}, \cdots, \beta_k^{2(-m)}$. Subsequently, we compute the areas under the Betti plots by discretizing the integral. The area differences between the groups are then tested using the Wilcoxon rank-sum test, which is a nonparametric test on median differences (Gibbons and Chakraborti, 2011). Using the jackknife resampling, we can avoid using the permutation test. For the permutation test to converge for our data set, it requires tens of thousands of permutations, and it is really time consuming even with the proposed time-saving soft-thresholding method. The proposed method takes about a minute of computation in a desktop but tens of thousands of permutations will take about seven days of computation. The MATLAB codes for constructing network filtration and barcodes and performing statical inference based on the jackknife resampling is provided.[1]

Comparisons between Betti plots can be also done by measuring the distance between the Betti plots. Given two networks $\mathcal{X}^1 = (V, w^1)$ and $\mathcal{X}^2 = (V, w^2)$, the Kolmogorov Smirnov (KS) distance between $\mathcal{X}^1$ and $\mathcal{X}^2$ is defined as (Chung et al., 2013, 2015a; Lee et al., 2017)

$$D_{KS}(\mathcal{X}^1, \mathcal{X}^2) = \sup_{\epsilon \geq 0} \left| f(\mathcal{X}^1_\epsilon) - f(\mathcal{X}^2_\epsilon) \right|$$

using monotone function f. The distance D_{KS} is motivated by the KS test for determining the equivalence of two cumulative distribution functions (Böhm and Hornik, 2010; Gibbons and Chakraborti, 2011; Chung et al., 2017a).

[1] http://brainimaging.waisman.wisc.edu/~chung/barcodes/

The distance D_{KS} can be discretely approximated using the finite number of filtrations:

$$D_q = \sup_{1 \le j \le q} \left| f(\mathcal{X}^1_{\epsilon_j}) - f(\mathcal{X}^2_{\epsilon_j}) \right|.$$

If we choose enough number of q such that ϵ_j are all the sorted edge weights, then

$$D_{KS}(\mathcal{X}^1, \mathcal{X}^2) = D_q$$

(Chung et al., 2017a). This is possible since there are only up to $p(p-1)/2$ number of unique edges in a graph with p nodes and the monotone function increases discretely but *not continuously*. In practice, ϵ_j may be chosen uniformly or a divide-and-conquer strategy can be used to adaptively grid the filtration values. We will study the inference procedure using D_q in a latter chapter.

7.4.2 The Effect of Node Numbers on Betti Numbers

Depending on the number of nodes, the parameters of graphs vary considerably up to 95% and the resulting statistical results will change substantially (Gong et al., 2009; Fornito et al., 2010; Zalesky et al., 2010). On the other hand, Betti plots are very robust under the change of node size since the plots monotonically change by the increment of up to one. In a structural covariate study on the white matter boundary (Chung et al., 2015a), for the node sizes between 548 and 1,856 (0.3% and 1% of original 189,536 mesh vertices), the choice of node size did not affect the pattern of graph filtrations, the shape of Betti plots, or the subsequent statistical results significantly. For example, the graph filtration on 1,856 nodes shows a similar pattern of dense connections for the maltreated children (Figure 7.8). The resulting Betti plots also show a similar pattern of the group separation. The statistical results are also somewhat consistent. For both the Jacobian determinant and fractional anisotropy (FA) values, the group differences in Betti plots obtained from sparse correlations and covariances are all statistically significant (p-value < 0.001) in both 548 and 1,856 nodes except one case. For the case of the 548 nodes' covariance on FA values, we did not detect any group differences at 0.01 level (p-value = 0.043). On the other hand, we detected the group difference for the 1,856 nodes case at 0.001 level.

8
Diffusions on Graphs

In brain imaging, the image acquisition and processing processes themselves are likely to introduce noise to the images. It is therefore imperative to reduce the noise while preserving the geometric details of the anatomical structures for various applications. Diffusion equations have been widely used in brain imaging as a form of noise reduction motivated by Perona and Malik (1990). Numerous diffusion-based techniques have been developed in image processing (Sochen et al., 1998; Tang et al., 1999; Taubin, 2000; Andrade et al., 2001; Chung et al., 2001a, 2003a, 2005b; Malladi and Ravve, 2002; Cachia et al., 2003a,b; Chung and Taylor, 2004; Joshi et al., 2009).

The direct application of Gaussian kernel smoothing tends to cause various numerical issues in irregular domains with boundaries. For example, if one uses large bandwidth in kernel smoothing in a cortical bounded region, the smoothing will blur signals across boundaries. So in kernel smoothing and regression literature, various ad hoc procedures were introduced to remedy the boundary effect. However, the most natural, straightforward way to smooth images in irregular domains with boundaries is to formulate the problem as boundary value problems using partial differential equations.

8.1 Diffusion as a Cauchy Problem

Consider $\mathcal{M} \in \mathbb{R}^d$ to be a compact differentiable manifold. Let $L^2(\mathcal{M})$ be the space of square integrable functions in $\mathcal{M}$ with inner product

$$\langle g_1, g_2 \rangle = \int_{\mathcal{M}} g_1(p) g_2(p) \, d\mu(p), \tag{8.1}$$

where μ is the Lebegue measure such that $\mu(\mathcal{M})$ is the total volume of $\mathcal{M}$. The norm $\| \cdot \|$ is defined as

$$\|g\| = \langle g, g \rangle^{1/2}.$$

The linear partial differential operator $\mathcal{L}$ is *self-adjoint* if

$$\langle g_1, \mathcal{L}g_2 \rangle = \langle \mathcal{L}g_1, g_2 \rangle$$

for all $g_1, g_2 \in L^2(\mathcal{M})$. Then the eigenvalues λ_j and eigenfunctions ψ_j of the operator $\mathcal{L}$ are obtained by solving

$$\mathcal{L}\psi_j = \lambda_j \psi_j. \tag{8.2}$$

Often (8.2) is written as

$$\mathcal{L}\psi_j = -\lambda_j \psi_j$$

so care should be taken in assigning the sign of eigenvalues.

Theorem 8.1 *The eigenfunctions ψ_j are orthonormal.*

Proof. Note $\langle \psi_i, \mathcal{L}\psi_j \rangle = \lambda_j \langle \psi_i, \psi_j \rangle$. On the other hand, $\langle \mathcal{L}\psi_i, \psi_j \rangle = \lambda_i \langle \psi_i, \psi_j \rangle$. Thus

$$(\lambda_i - \lambda_j)\langle \psi_i, \psi_j \rangle = 0.$$

For any $\lambda_i \neq \lambda_j$, $\langle \psi_i, \psi_j \rangle = 0$, orthogonal. For ψ_j to be orthonormal, we need $\langle \psi_j, \psi_j \rangle = 1$. This is simply done by absorbing the constant multiple into ψ_j. Thus, $\{\psi_j\}$ is orthonormal. □

In fact, ψ_j is the basis in $L^2(\mathcal{M})$. Consider 1D eigenfunction problem

$$\frac{\partial^2}{\partial x^2}\psi_j(x) = -\lambda_j \psi_j(x)$$

in interval $[-l, l]$. We can easily check that

$$\psi_{1j} = \cos\left(\frac{j\pi x}{l}\right), \ \psi_{2j} = \sin\left(\frac{j\pi x}{l}]\right), \ j = 1, 2, \cdots$$

are eigenfunctions corresponding to eigenvalue $\lambda_j = \left(\frac{j\pi}{l}\right)^2$. Also $\psi_{10} = 1$ is the trivial first eigenfunction corresponding to $\lambda_0 = 0$. The multiplicity of eigenfunctions is caused by the symmetric of interval $[-l, l]$. Based on trigonometric formula, we can show that the eigenfunctions are orthogonal:

$$\int_{-l}^{l} \psi_{1i}(x)\psi_{1j}(x)\, dx = 0 \text{ if } i \neq j$$

$$\int_{-l}^{l} \psi_{2i}(x)\psi_{2j}(x)\, dx = 0 \text{ if } i \neq j$$

$$\int_{-l}^{l} \psi_{1i}(x)\psi_{2j}(x)\, dx = 0 \text{ for any } i, j$$

From $\psi_{1j}^2(x) + \psi_{2j}^2(x) = 1$ and due to symmetry,

$$\int_{-l}^{l} \psi_{1j}^2(x)\, dx = \int_{-l}^{l} \psi_{2j}^2(x)\, dx = l.$$

Thus

$$\begin{aligned}
\psi_{10} &= \frac{1}{\sqrt{2l}},\\
\psi_{1j} &= \frac{1}{\sqrt{l}} \cos\left(\frac{j\pi x}{l}\right),\\
\psi_{2j} &= \frac{1}{\sqrt{l}} \sin\left(\frac{j\pi x}{l}\right),\ j = 1, 2, \cdots
\end{aligned}$$

are orthonormal bases in $[-l, l]$.

Consider a Cauchy problem of the following form:

$$\frac{\partial g}{\partial t}(p,t) + \mathcal{L}g(p,t) = 0, g(p,t=0) = f(p), \tag{8.3}$$

where t is the time variable and p is the spatial variable.

The initial functional data $f(p)$ can be further stochastically modeled as

$$f(p) = \nu(p) + \epsilon(p), \tag{8.4}$$

where ϵ is a stochastic noise modeled as a zero-mean Gaussian random field, i.e., $\mathbb{E}\epsilon(p) = 0$ at each point p and ν is the unknown signal to be estimated. Partial differential equation (PDE) (8.3) diffuses noisy initial data f over time and estimate the unknown signal ν as a solution. Diffusion time t controls the amount of smoothing and will be termed as the *bandwidth*. The unique solution to equation (8.3) is given as follows. This is a heuristic proof, and more rigorous proof is given later.

Theorem 8.2 *For the self-adjoint linear differential operator $\mathcal{L}$, the unique solution of the Cauchy problem*

$$\frac{\partial g}{\partial t}(p,t) + \mathcal{L}g(p,t) = 0, g(p,t=0) = f(p) \tag{8.5}$$

is given by

$$g(p,t) = \sum_{j=0}^{\infty} e^{-\lambda_j t} \langle f, \psi_j \rangle \psi_j(p). \tag{8.6}$$

Proof. For each fixed t, since $g \in L^2(\mathcal{M})$, g has expansion

$$g(p,t) = \sum_{j=0}^{\infty} c_j(t)\psi_j(p). \tag{8.7}$$

Substitute equation (8.7) into (8.5). Then we obtain

$$\frac{\partial}{\partial t} c_j(t) + \lambda_j c_j(t) = 0 \tag{8.8}$$

for all j. The solution of equation (8.8) is given by

$$c_j(t) = b_j e^{-\lambda_j t}.$$

So we have solution

$$g(p,t) = \sum_{j=0}^{\infty} b_j e^{-\lambda_j t} \psi_j(p).$$

At $t = 0$, we have

$$g(p,0) = \sum_{j=0}^{\infty} b_j \psi_j(p) = f(p).$$

The coefficients b_j must be the Fourier coefficients $\langle f, \psi_j \rangle$ and they are uniquely determined. □

The implication of Theorem 8.2 is obvious. The solution decreases exponentially as time t increases and smooths out high spatial frequency noise much faster than low-frequency noise. This is the basis of many of PDE-based image smoothing methods. PDEs involving self-adjoint linear partial differential operators such as the Laplace–Beltrami operator or iterated Laplacian have been widely used in medical image analysis as a way to smooth either scalar or vector data along anatomical boundaries (Andrade et al., 2001; Chung et al., 2003a; Bulow, 2004). These methods directly solve PDE using standard numerical techniques such as the finite difference method (FDM) or the finite element method (FEM). The main shortcoming of solving PDE using FDM or FEM is the numerical instability and the complexity of setting up the numerical scheme. The analytic approach called weighted Fourier series (WFS) differs from these previous methods in such a way that we only need to estimate the Fourier coefficients in a hierarchical fashion to solve PDE.

Example 8.1 *Consider 1D differential operator $\mathcal{L} = \frac{\partial^2}{\partial x^2}$. The corresponding Cauchy problem is 1D diffusion equation*

$$\frac{\partial g}{\partial t}(p,t) + \frac{\partial^2 g}{\partial x^2}(p,t) = 0, g(p,t=0) = f(p), p \in [-l,l].$$

Then the solution of this problem is given by Theorem 8.2:

$$g(p,t) = a_0\psi_{10} + \sum_{j=1}^{\infty} a_j e^{-\lambda_j t} \psi_{1j}(p) + b_j e^{-\lambda_j t} \psi_{2j}(p),$$

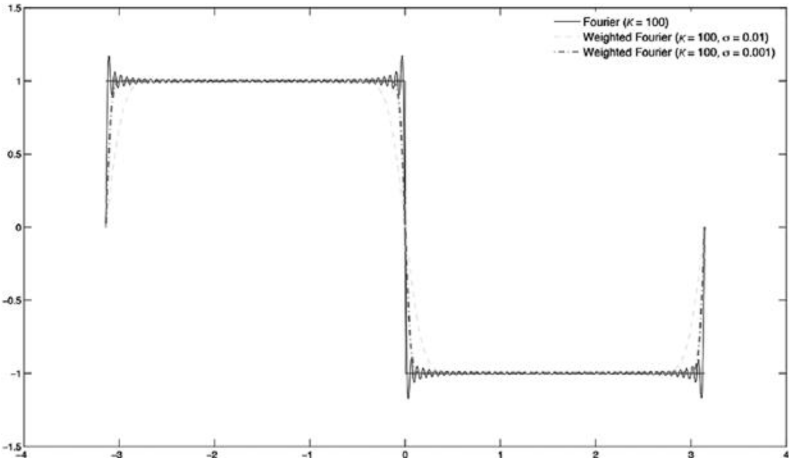


Figure 8.1 Reduction of Gibbs phenomenon in weighted Fourier series (WFS). The figure was generated by Yuan Wang of University of Wisconsin–Madison (Wang et al., 2018).

where

$$a_0 = \frac{1}{\sqrt{2l}} \int_{-l}^{l} f(p)\, dp,$$

$$a_j = \frac{1}{\sqrt{l}} \int_{-l}^{l} f(p) \cos\left(\frac{j\pi x}{l}\right) dp,$$

$$b_j = \frac{1}{\sqrt{l}} \int_{-l}^{l} f(p) \sin\left(\frac{j\pi x}{l}\right) dp$$

for $j = 1, 2, \cdots$ *(see Figure 8.1)*

8.2 Finite Difference Method

One way of solving diffusion equations numerically is to use finite differences. We will discuss how to differentiate images. There are numerous techniques for differentiation proposed in the literature. We start with simple example of image differentiation in 2D image slices. Consider image intensity $f(x, y)$ defined on a regular grid, i.e., $(x, y) \in \mathbb{Z}^2$. Assume the pixel size is δx and δy in the x- and y-directions. The partial derivative along the x-direction of image f is approximated by the finite difference:

$$\frac{\partial f}{\partial x}(x, y) = \frac{f(x + \delta x, y) - f(x, y)}{\delta x}.$$

The partial derivative along the y-direction of image f is approximated similarly. $\frac{\partial f}{\partial x}(x,y)$ and $\frac{\partial f}{\partial y}(x,y)$ are called the *first-order derivatives*. Then the *second-order derivatives* are defined by taking the finite difference twice:

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2}(x,y) &= \left[\frac{f(x+\delta x,y)-f(x,y)}{\delta x} - \frac{f(x,y)-f(x-\delta x,y)}{\delta x}\right]\Big/\delta x\\ &= \frac{f(x+\delta x,y)-2f(x,y)+f(x-\delta x,y)}{\delta x^2}\end{aligned}$$

Similarly, we also have

$$\frac{\partial^2 f}{\partial y^2}(x,y) = \frac{f(x,y+\delta y)-2f(x,y)+f(x,y-\delta y)}{\delta x^2}.$$

Other partial derivatives such as $\frac{\partial^2 f}{\partial x \partial y}$ are computed similarly.

8.2.1 1D Diffusion by Finite Difference

Let us implement 1D version of diffusion equations (Figure 8.2). Suppose we have a smooth function $f(x,t)$ that is a function of position $x \in \mathbb{R}$ and time $t \in \mathbb{R}^+$. 1D isotropic heat equation is then defined as

$$\frac{\partial f}{\partial t} = \frac{d^2 f}{dx^2} \tag{8.9}$$

with initial condition $f(x,t=0) = g(x)$. Differential equation (8.9) is then discretized as

$$f(x,t+\delta t) = f(x,t) + \delta t \frac{d^2 f}{dx^2}(x,y). \tag{8.10}$$

With $t_k = k\delta t$ and starting from $t = 0$, (8.10) can be written as

$$f(x,t_{k+1}) = f(x,t_k) + \delta t \frac{f(x+\delta x,t_k) - 2f(x,t_k) + f(x-\delta x,t_k)}{\delta x^2}.$$

The preceding finite difference gives the solution at time t_{k+1}. To obtain the solution at any time, it is necessary to keep iterating many times with very small δt. If δt is too small, the computation is slow. If it is too large, the finite difference will diverge. Then the problem is finding the largest δt that guarantees the convergence.

Discrete maximum principle. Since the diffusion smoothing and kernel smoothing are equivalent, The diffused signal $f(x,t_{k+1})$ must be bounded

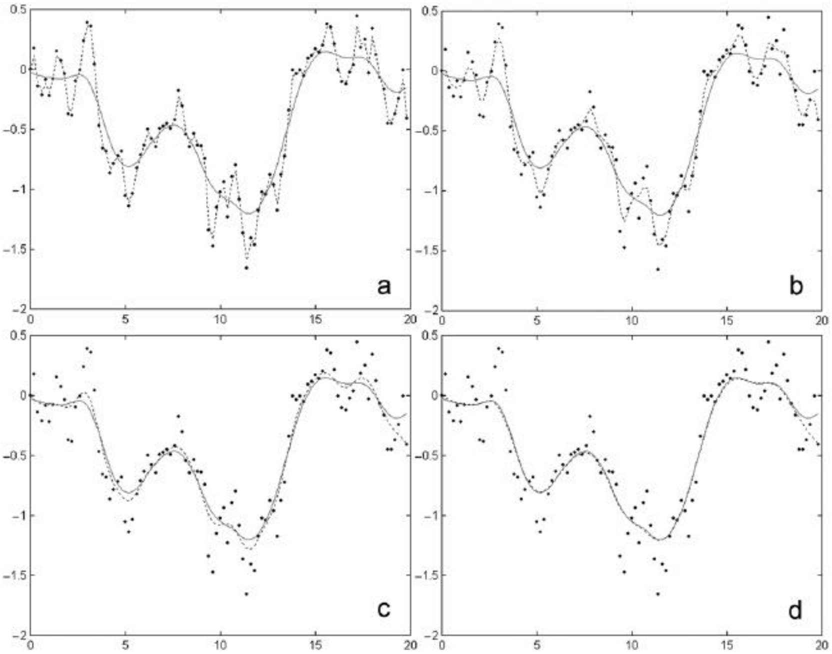


Figure 8.2 Gaussian kernel smoothing (solid line) and diffusion smoothing (dotted line) (a) before diffusion, (b) after 0.05 seconds (five iterations), (c) after 0.25 seconds (25 iterations), and (d) after 0.5 seconds (50 iterations).

by the minimum and the maximum of signal (Chung et al., 2003c). Let x_{i-1}, x_i, x_{i+1} be some points with gap δx.

$$\begin{aligned} f(x_i, t_{k+1}) &= f(x_i, t_k) + \delta t \frac{d^2 f}{dx^2}(x_i, t_k) \\ &\leq \max \big[f(x_{i-1}, t_k), f(x_i, t_k), f(x_{i+1}, t_k) \big]. \end{aligned}$$

Similarly, we can bound it as follows. Thus, the time step should be bounded by

$$\delta t \leq \max \left[\left| \frac{f(x_{i-1}, t_j) - f(x_i, t_j)}{\frac{d^2 f}{dx^2}} \right|, \left| \frac{f(x_{i+1}, t_j) - f(x_i, t_j)}{\frac{d^2 f}{dx^2}} \right| \right].$$

8.2.2 Diffusion in *n*-Dimensional Grid

In 2D, let (x_i, y_i) be pixels around (x, y) including (x, y) itself. Then using the four-neighbor scheme, the Laplacian of $f(x, y)$ can be written as

$$\Delta f(x,y) = \sum_{i,j} w_{ij} f(x_i, y_i),$$

where the Laplacian matrix is given by

$$(w_{ij}) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Note that $\sum_{ij} w_{ij} = 1$.

Extending it further, consider nD. Let $x = (x_1, x_2, \cdots, x_n)$ be the coordinates in $\mathbb{R}^n$. Laplacian Δ in $\mathbb{R}^n$ is defined as

$$\Delta f = \frac{\partial^2 f}{\partial x_1^2} + \cdots + \frac{\partial^2 f}{\partial x_n^2}.$$

Assume we have a n-dimensional hypercube grid of size is 1. Then we have

$$\begin{aligned} \Delta f(x) = f(x_1 \pm 1, \cdots, x_n) + \cdots + f(x_1, \cdots, x_n \pm 1) \\ -2nf(x,y). \end{aligned}$$

This uses $2n$ closest neighbors of voxel x to approximate the Laplacian.

It is also possible to incorporate 2^n corners $(x_1 \pm 1, \cdots, x_n \pm 1)$ along with the $2n$ closest neighbors for a better approximation of the Laplacian. In particular, in 2D we can obtain a more accurate finite difference formula for eight-neighbor Laplacian:

$$(w_{ij}) = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Based on the estimation Laplacian on a discrete grid, diffusion equation

$$\frac{\partial f}{\partial t} = \Delta f$$

is discretized as

$$f(x, t_{k+1}) = f(x, t_k) + \delta t \sum_{i,j} w_{ij} f(x_i, y_i), \tag{8.11}$$

with $t_k = k\delta t$ and starting from $t_1 = 0$. From (8.11), we can see that the diffusion equation is solved by iteratively applying convolution with weights w_{ij}. In fact, it can be shown that the solution of diffusion is given by kernel smoothing.

8.3 Laplacian on Planner Graphs

In a previous section, we showed how to estimate the Laplacian in a regular grid. Now we show how to estimate Laplacian in an irregular grid such as graphs and polygonal surfaces in $\mathbb{R}^2$. The question is how one can estimate Laplacian or any other differential operators on a graph. Assume we have observations Y_i at each point p_i, which is assumed to follow the additive model

$$Y_i = \mu(p_i) + \epsilon(p_i),\ \ p_i \in \mathbb{R}^2$$

where μ is a smooth continuous function and ϵ is a zero mean Gaussian random field. We want to estimate at some node p_i on a graph:

$$\Delta\mu(p_0) = \frac{\partial^2 \mu}{\partial x^2}\Big|_{p_0} + \frac{\partial^2 \mu}{\partial y^2}\Big|_{p_0}.$$

Unfortunately, the geometry of the graph forbids direct application of the finite difference scheme. To answer this problem, one requires the FEM (Chung, 2001). However, we can use a more elementary technique called *polynomial regression*.

Let $p_i = (x_i, y_i)$ be the coordinates of the vertices of the graph or polygonal surface. Let p_i be the neighboring vertices of p_0. We estimate the Laplacian at p_0 by fitting a quadratic polynomial of the form

$$\mu(u,v) = \beta_0 + \beta_1 u + \beta_2 v + \beta_3 u^2 + \beta_4 uv + \beta_5 v^2. \tag{8.12}$$

We are basically assuming the unknown signal μ to be the quadratic form (8.12). Then the parameters β_i are estimated by solving the normal equation:

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 y_i + \beta_3 x_i^2 + \beta_4 x_i y_i + \beta_5 y_i^2 \tag{8.13}$$

for all p_i that is neighboring p_0. For simplicity, we may assume p_0 is translated to the origin, i.e., $x_0 = 0, y_0 = 0$.

Let $Y = (Y_1, \cdots, Y_m)^\top$, $\beta = (\beta_0, \cdots, \beta_5)^\top$ and design matrix

$$\mathbb{X} = \begin{pmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2 y_2 & y_2^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_m & y_m & x_m^2 & x_m y_m & y_m^2 \end{pmatrix}.$$

Then we have the following matrix equation

$$Y = \mathbb{X}\beta.$$

The unknown coefficients vector β is estimated by the usual least squares method:

$$\widehat{\beta} = (\widehat{\beta_0}, \ldots, \widehat{\beta_5})^\top = (\mathbb{X}^\top \mathbb{X})^{-}\mathbb{X}^\top Y,$$

where $^-$ denotes generalized inverse, which can be obtained through the singular value decomposition (SVD). Note that $\mathbb{X}^\top\mathbb{X}$ is nonsingular if $m < 6$. In MATLAB, `pinv` can be used to compute the generalized inverse, which is often called the *pseudoinverse*.

The generalized inverse often used is that of Moore–Penrose. It is usually defined as matrix $\mathbb{X}$ satisfying four conditions:

$$\mathbb{X}\mathbb{X}^-\mathbb{X} = \mathbb{X},\ \mathbb{X}^-\mathbb{X}\mathbb{X}^- = \mathbb{X}^-,$$

$$(\mathbb{X}\mathbb{X}^-)^\top = \mathbb{X}\mathbb{X}^-,\ (\mathbb{X}^-\mathbb{X})^\top = \mathbb{X}^-\mathbb{X}.$$

Let $\mathbb{X}$ be $m \times p$ matrix with $m \geq p$. Then the SVD of $\mathbb{X}$ is

$$\mathbb{X} = UDV^\top,$$

where $U_{m\times p}$ has orthonormal columns, $V_{p\times p}$ is orthogonal, and $D_{p\times p} = Diag(d_1, \cdots, d_p)$ is diagonal with nonnegative elements. Let

$$D^- = Diag(d_1^-, \cdots, d_p^-),$$

where $d_i^- = 1/d_i$ if $d_i \neq 0$ and $d_i^- = 0$ if $d_i = 0$. Then it can be shown that the Moore–Penrose generalized inverse is given by

$$\mathbb{X}^- = VD^-U^\top.$$

Once we have estimated the parameter vector β, the Laplacian is

$$\Delta\mu(p_0) = 2\widehat{\beta}_3 + 2\widehat{\beta}_5.$$

8.4 Graph Laplacian

Now we generalize volumetric Laplacian in previous sections to graphs. Let $G = (V, E)$ be a graph with node set V and edge set E. We will simply index the node set as $V = \{1, 2, \cdots, p\}$. If two nodes i and j form an edge, we denote it as $i \sim j$. Let $W = (w_{ij})$ be the edge weight. The adjacency matrix of G is often used as the edge weight. Various forms of graph Laplacian have been proposed (Chung and Yau, 1997), but the most often used standard form $L = (l_{ij})$ is given by

$$l_{ij} = \left(\begin{array}{cc} -w_{ij}, & i \sim j \\ \sum_{i\neq j} w_{ij}, & i = j \\ 0, & \text{otherwise} \end{array} \right.$$

Often it is defined with the sign reversed such that

$$l_{ij} = \left(\begin{array}{cc} w_{ij}, & i \sim j \\ -\sum_{i \neq j} w_{ij}, & i = j \\ 0, & \text{otherwise} \end{array} \right.$$

The graph Laplacian L can then be written as

$$L = D - W,$$

where $D = (d_{ij})$ is the diagonal matrix with $d_{ii} = \sum_{j=1}^{n} w_{ij}$. Here, we will simply use the adjacency matrix so that the edge weights w_{ij} are either 0 or 1. In MATLAB, Laplacian `L` is simply computed from the adjacency matrix `adj`:

```
n=size(adj,1);
adjsparse = sparse(n,n);
adjsparse(find(adj))=1;
L=sparse(n,n);
GL = inline('diag(sum(W))-W');
L = GL(adjsparse);
```

We use the sparse matrix format to reduce the memory burden for large-scale computation.

Theorem 8.3 *Graph Laplacian L is nonnegative definite.*

Proof. The proof is based on factoring Laplacian L using incidence matrix ∇ such that $L = \nabla^\top \nabla$. Such factorization always yields nonnegative definite matrices. Very often L is nonnegative definite in practice if it is too sparse (Figure 8.3).

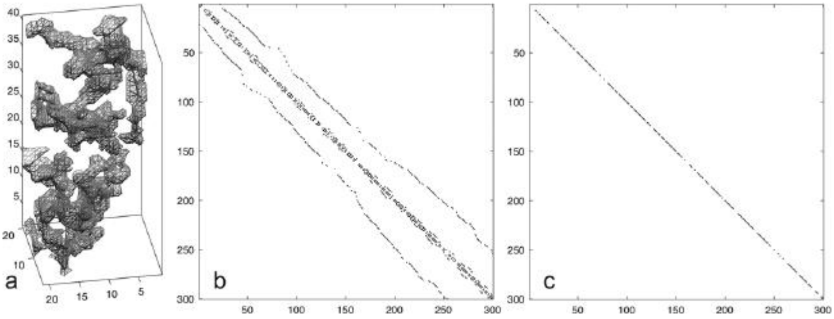


Figure 8.3 (a) Part of lung blood vessel obtained from CT. (b) Adjacency matrix obtained from four-neighbor connectivity. (c) Laplace matrix obtained from the adjacency matrix.

Theorem 8.4 *For graph Laplacian L, $L+\alpha I$ is positive definite for any $\alpha > 0$.*

Proof. Since L is nonnegative definite, we have

$$x^\top Lx \geq 0.$$

Then it follows that

$$x^\top (L+\alpha I)x = x^\top Lx + \alpha x^\top x > 0$$

for any $\alpha > 0$ and $x \neq 0$. □

Unlike the continuous Laplace–Beltrami operators that may have a possibly infinite number of eigenfunctions, we have up to p number of eigenvectors $\psi_1, \psi_2, \cdots, \psi_p$ satisfying

$$L\psi_j = \lambda_j \psi_j \tag{8.14}$$

with (Figure 8.4)

$$0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_p.$$

The eigenvectors are orthonormal, i.e.,

$$\psi_i^\mathsf{T} \psi_j = \delta_{ij},$$

the Kroneker's delta. The first eigenvector is trivially given as $\psi_1 = \mathbf{1}/\sqrt{p}$ with $\mathbf{1} = (1, 1, \cdots, 1)^\mathsf{T}$.

All other higher-order eigenvalues and eigenvectors are unknown analytically and have to be computed numerically (Figure 8.4). Using the eigenvalues and eigenvectors, the graph Laplacian can be decomposed spectrally. From (8.14),

$$L\Psi = \Psi\Lambda, \tag{8.15}$$

where $\Psi = [\psi_1, \cdots, \psi_p]$ and Λ is the diagonal matrix with entries $\lambda_1, \cdots, \lambda_p$. Since Ψ is an orthogonal matrix,

$$\Psi\Psi^\mathsf{T} = \Psi^\mathsf{T}\Psi = \sum_{j=1}^{p} \psi_j \psi_j^\mathsf{T} = I_p,$$

the identify matrix of size p. Then (8.15) is written as

$$L = \Psi\Lambda\Psi^\mathsf{T} = \sum_{j=1}^{p} \lambda_j \psi_j \psi_j^\mathsf{T}.$$

This is the restatement of the SVD for Laplacian.

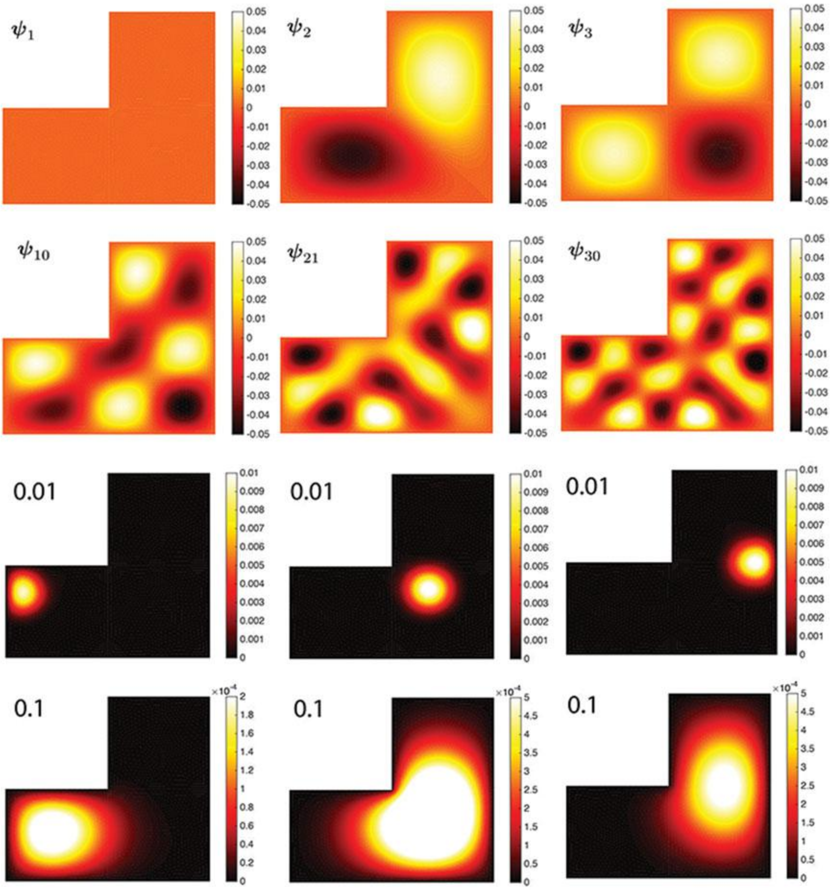


Figure 8.4 Top: first few eigenvectors of the Laplacian in an L-shaped domain. Bottom: heat kernel with bandwidths $\sigma = 0.01, 0.1$. We have used degree 70 expansions, but the shape is almost identical if we use higher-degree expansions. The heat kernel is a probability distribution that follows the shape of the L-shaped domain.

For measurement vector $f = (f_1, \cdots, f_p)^\top$ observed at the p nodes, the discrete Fourier series expansion is given by

$$f = \sum_{j=1}^{n} \tilde{f}_j \psi_j,$$

where $\tilde{f}_j = f^\top \psi_j = \psi_j^\top f$ are Fourier coefficients.

8.5 Fiedler Vectors

The connection between the eigenfunctions of continuous and discrete Laplacians have been well established by many authors (Gladwell and Zhu, 2002; Tlusty, 2007). Many properties of eigenfunctions of Laplace–Beltrami operators have discrete analogs. The second eigenfunction of the graph Laplacian is called the Fiedler vector and it has been studied in connection to the graph and mesh manipulation, manifold learning, and the minimum linear arrangement problem (Fiedler, 1973; Lévy, 2006; Ham et al., 2004, 2005).

Let $G = \{V, E\}$ be the graph with the vertex set V and the edge set E. We will simply index the node set as $V = \{1, 2, \cdots, n\}$. If two nodes i and j form an edge, we denote it as $i \sim j$. The edge weight between i and j is denoted as w_{ij}. For a measurement vector $\mathbf{f} = (f_1, \cdots, f_n)^\top$ observed at the n nodes, the discrete Dirichlet energy is given by

$$\mathcal{E}(\mathbf{f}) = \mathbf{f}^\top L\mathbf{f} = \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 = \sum_{i\sim j} w_{ij}(f_i - f_j)^2. \tag{8.16}$$

The discrete Dirichlet energy (8.16) is also called the linear placement cost in the minimum linear arrangement problem (Koren and Harel, 2002). Fielder vector $\mathbf{f}$ evaluated at n nodes is obtained as the minimizer of the quadratic polynomial:

$$\min_f \mathcal{E}(f)$$

subject to the quadratic constraint

$$\|\mathbf{f}\|^2 = \mathbf{f}^\top\mathbf{f} = \sum_i f_i^2 = 1. \tag{8.17}$$

The solution can be interpreted as the kernel principal components of a Gram matrix given by the generalized inverse of L (Ham et al., 2004, 2005). Since the eigenvector ψ_1 of Laplacian is orthonormal with eigenvector ψ_0, which is constant, we also have an additional constraint:

$$\sum_i f_i = 0. \tag{8.18}$$

This optimization problem was first introduced for the minimum linear arrangement problem in 1970s (Hall, 1970; Koren and Harel, 2002). The optimization can be solved using the Lagrange multiplier as follows (Holzrichter and Oliveira, 1999).

Let g be the constraint (8.17) so that

$$g(\mathbf{f}) = \mathbf{f}^\top\mathbf{f} - 1 = 0.$$

Then the constrainted minimum should satisfy

$$\nabla \mathcal{E} - \mu \nabla g = 0, \tag{8.19}$$

where μ is the Lagrange multiplier. (8.19) can be written as

$$2L\mathbf{f} - \mu\mathbf{f} = 0 \tag{8.20}$$

Hence, $\mathbf{f}$ must be the eigenvector of L and $\mu/2$ is the corresponding eigenvalue. By multiplying $\mathbf{f}^\top$ on the both sides of (8.20), we have

$$2\mathbf{f}^\top L\mathbf{f} = \mu\mathbf{f}^\top\mathbf{f} = \mu.$$

Since we are minimizing $\mathbf{f}^\top L\mathbf{f}$, $\mu/2$ should be the second eigenvalue λ_1.

In most literature (Holzrichter and Oliveira, 1999), the condition $\sum_i f_i = 0$ is incorrectly stated as a necessary constraint for the Fiedler vector. However, the constraint $\sum_i f_i = 0$ is not really needed in minimizing the Dirichlet energy. This can be further seen from introducing a new constraint

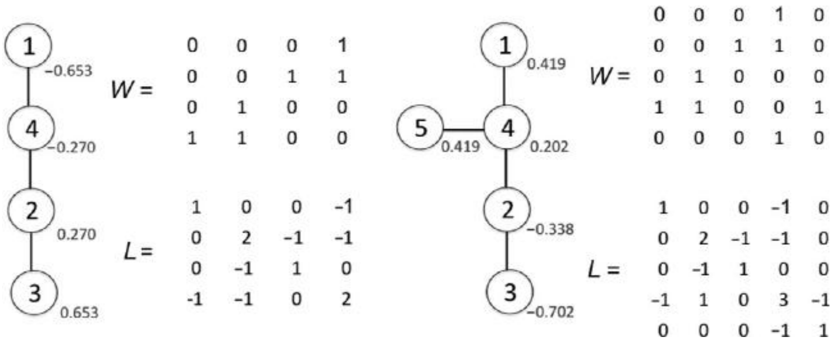
$$h(\mathbf{f}) = \mathbf{e}^\top\mathbf{f} = \sum_i f_i = 0,$$

where $\mathbf{e} = (1, \cdots, 1)^\top$.

The constraint (8.17) and (8.18) forces ψ_1 to have at least two differing *sign domains* in which ψ_1 has one sign. But it is unclear how many differing sign domains ψ_1 can possibly have. The upper bound is given by Courant's nodal line theorem (Courant and Hilbert, 1953; Gladwell and Zhu, 2002; Tlusty, 2007). The *nodal set* of eigenvector ψ_i is defined as the zero-level set $\psi_i(p) = 0$. Courant's nodal line theorem states that the nodal set of the ith eigenvector ψ_i divides the graph into no more than i sign domains. Hence, the second eigenvector has exactly two disjoint sign domains. At the positive sign domain, we have the global maximum and at the negative sign domain, we have the global minimum. This property is illustrated in Figure 8.5. However, it is unclear where the global maximum and minimum are located. The concept of tightness is useful in determining the location.

Definition 8.1 *For a function* $\mathbf{f}$ *defined on vertex set* V *of* G*, let* G_s^- *be the subgraph of* G *induced by the vertex set* $V_s^- = \{i \in V | f_i < s\}$*. Let* G_s^+ *be the subgraph of* G *induced by the vertex set* $V_s^+ = \{i \in V | f_i > s\}$*. For any* s*, if* G_s^- *and* G_s^+ *are either connected or empty, then* $\mathbf{f}$ *is tight (Tlusty, 2007).*

When $s = 0$, G_0^+ and G_0^- are sign graphs. If we relax the condition so that G_s^+ contains nodes satisfying $f_i \geq s$, we have weak sign graphs. It can be shown that the second eigenvector on a graph with maximal degree 2 (cycle or path) is tight (Tlusty, 2007). Figure 8.6 shows an example of a path with



$$W = \begin{matrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{matrix} \qquad L = \begin{matrix} 1 & 0 & 0 & -1 \\ 0 & 2 & -1 & -1 \\ 0 & -1 & 1 & 0 \\ -1 & -1 & 0 & 2 \end{matrix}$$

$$W = \begin{matrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{matrix} \qquad L = \begin{matrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 2 & -1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 3 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{matrix}$$

Figure 8.5 A weighted graph with weights W and the graph Laplacian L. The weights are simply the adjacency matrix. The second eigenvector ψ_1 is given as numbers beside nodes. Left: this example is given in Hall (1970). The maximum geodesic distance is obtained between the nodes 1 and 3, which are also hot and cold spots. Right: there are two hot spots 1 and 5, which correspond to two maximal geodesic paths 1-4-2-3 and 5-4-2-3 (Chung et al., 2011b).

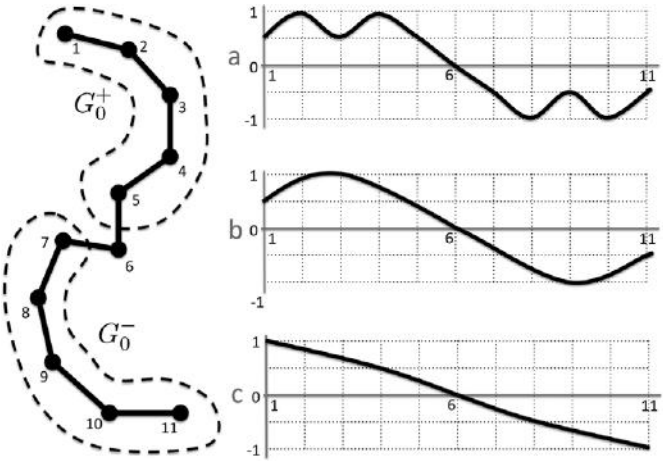


Figure 8.6 A path with positive (G_0^+) and negative (G_0^-) sign domains. Due to symmetry, the possible eigenfunction ψ_1 has to be an odd function.

11 nodes. Among three candidates for the second eigenfunction, (a) and (b), are not tight while (c) is. Note that the candidate function (a) have two disjoint components for $G_{0.5}^+$ so it cannot be tight. In order to be tight, the second eigenfunction cannot have a positive minimum or a negative maximum at the interior vertex in the graph (Gladwell and Zhu, 2002). This implies that the second eigenfunction must decrease monotonically from the positive to

negative sign domains as shown in (c). Therefore, the hot and cold spots must occur at the two end points, 1 and 11, which gives the maximum geodesic distance of 11.

For a cycle, the argument is similar except that a possible eigenfunction has to be periodic and tight, which forces the hot and cold spots to be located at the maximum distance apart. Due to the periodicity, we will have multiplicity of eigenvalues in the cycle. Although it is difficult to predict the location of maximum and minimums in general, the behavior of the second eigenfunction is predictable for an elongated graph; it provides an intrinsic geometric way of establishing natural coordinates.

8.6 Heat Kernel Smoothing on Graphs

Heat kernel smoothing was originally introduced in the context of filtering out cortical surface data defined on mesh vertices obtained from 3D medical images (Chung et al., 2005a,b). The formulation uses the tangent space projection in approximating the heat kernel by iteratively applying Gaussian kernel with smaller bandwidth. Recently proposed spectral formulation to heat kernel smoothing (Chung et al., 2015b) constructs the heat kernel analytically using the eigenfunctions of the Laplace–Beltrami (LB) operator, avoiding the need for the linear approximation used in (Chung et al., 2005b; Han et al., 2006). Since surface meshes are graphs, heat kernel smoothing can be used to smooth noisy data defined on network nodes.

Instead of the Laplace–Beltrami operator for cortical surface, graph Laplacian is used to construct the discrete version of heat kernel smoothing. The connection between the eigenfunctions of continuous and discrete Laplacians has been well established by several studies (Gladwell and Zhu, 2002; Tlusty, 2007). Although many have introduced the discrete version of heat kernels in computer vision and machine learning, they mainly used the heat kernels to compute shape descriptors or to define a multiscale metric (Belkin et al., 2006; de Goes et al., 2008; Sun et al., 2009; Bronstein and Kokkinos, 2010). These studies did not use the heat kernel in filtering out data on graphs. There have been significant developments in kernel methods in the machine learning community (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004; Nilsson et al., 2007; Steinke and Hein, 2008; Yger and Rakotomamonjy, 2011). However, the heat kernel has never been used in such frameworks. Most kernel methods in machine learning deal with the linear combination of kernels as a solution to penalized regressions. On the other hand, our kernel method does not have a penalized cost function.

8.6.1 Heat Kernel on Graphs

The *discrete heat kernel* K_σ is a positive definite symmetric matrix of size $p \times p$ given by

$$K_\sigma = \sum_{j=1}^{p} e^{-\lambda_j \sigma} \psi_j \psi_j^\top, \tag{8.21}$$

where σ is called the bandwidth of the kernel. Figure 8.4 displays heat kernel with different bandwidths at an L-shaped domain. Alternately, we can write (8.21) as

$$K_\sigma = \Psi e^{-\sigma\Lambda} \Psi^\top,$$

where $e^{-\sigma\Lambda}$ is the matrix logarithm of Λ. To see positive definiteness of the kernel, for any nonzero $x \in \mathbb{R}^p$,

$$\begin{aligned} x^\top K_\sigma x &= \sum_{j=1}^{p} e^{-\lambda_j \sigma} x^\top \psi_j \psi_j^\top x \\ &= \sum_{j=1}^{p} e^{-\lambda_j \sigma} (\psi_j^\top x)^2 > 0. \end{aligned}$$

When $\sigma = 0$, $K_0 = I_p$, the identity matrix. When $\sigma = \infty$, by interchanging the sum and the limit, we obtain

$$K_\infty = \psi_1 \psi_1^\top = \mathbf{1}\mathbf{1}^\top / p.$$

K_∞ is a degenerate case and the kernel is no longer positive definite. Other than these specific cases, the heat kernel is not analytically known in arbitrary graphs.

Heat kernel is doubly stochastic (Chung and Yau, 1997) so that

$$K_\sigma \mathbf{1} = \mathbf{1}, \ \mathbf{1}^\top K_\sigma = \mathbf{1}^\top.$$

Thus, K_σ is a probability distribution along columns or rows.

Just like the continuous counterpart, the discrete heat kernel is also multiscale and has the scale-space property. Note

$$\begin{aligned} K_\sigma^2 &= \sum_{i,j=1}^{p} e^{-(\lambda_i + \lambda_j)\sigma} \psi_i \psi_i^\top \psi_j \psi_j^\top \\ &= \sum_{j=1}^{p} e^{-2\lambda_j \sigma} \psi_j \psi_j^\top = K_{2\sigma}. \end{aligned}$$

We used the orthonormality of eigenvectors. Subsequently, we have

$$K_\sigma^n = K_{n\sigma}.$$

8.6.2 Heat Kernel Smoothing on Graphs

Discrete heat kernel smoothing of measurement vector f is then defined as convolution

$$K_\sigma * f = K_\sigma f = \sum_{j=0}^{p} e^{-\lambda_j \sigma} \tilde{f}_j \psi_j, \tag{8.22}$$

This is the discrete analog of the heat kernel smoothing first defined in (Chung et al., 2005b). In a discrete setting, the convolution $*$ is simply a matrix multiplication. Thus,

$$K_0 * f = f$$

and

$$K_\infty * f = \bar{f}\mathbf{1},$$

where $\bar{f} = \sum_{j=1}^{p} f_j/p$ is the mean of signal f over every node. When the bandwidth is zero, we are not smoothing data. As the bandwidth increases, the smoothed signal converges to the sample mean over all nodes.

Define the l-norm of a vector $f = (f_1, \cdots, f_p)^\top$ as

$$\parallel f \parallel_l = \Big(\sum_{j=1}^{p} |f_j|^l \Big)^{1/l}.$$

The matrix ∞-norm is defined as

$$\parallel f \parallel_\infty = \max_{1 \le j \le p} |f_j|.$$

Theorem 8.5 *Heat kernel smoothing is a contraction mapping with respect to the lth norm, i.e.,*

$$\|K_\sigma * f\|_l^l \le \|f\|_l^l.$$

Proof. Let kernel matrix $K_\sigma = (k_{ij})$. Then we have inequality

$$\|K_\sigma * f\|_l^l = \sum_{i=1}^{p} \sum_{j=1}^{p} |k_{ij} f_j|^l \le \sum_{j=1}^{p} |f_j|^l.$$

We used Jensen's inequality and doubly stochastic property of the heat kernel. Similarly, we can show that heat kernel smoothing is a contraction mapping with respect to the ∞-norm as well.

Theorem 8.5 shows that heat kernel smoothing contracts the overall size of data. This fact can be used to skeltonize the blood vessel trees.

8.6.3 Statistical Properties

Often observed noisy data f on graphs is smoothed with heat kernel K_σ to increase the signal-to-noise ratio (SNR) and increases the statistical sensitivity (Chung et al., 2015b). We are interested in knowing how heat kernel smoothing will affect the statistical properties of smoothed data.

Consider the following addictive noise model:

$$f = \mu + e, \tag{8.23}$$

where μ is an unknown signal and ϵ is zero mean noise. Let $e = (e_1, \cdots, e_p)^\top$. Denote $\mathbb{E}$ as expectation and $\mathbb{V}$ as covariance. It is natural to assume that the noise variabilities at different nodes are identical, i.e.,

$$\mathbb{E}e_1^2 = \mathbb{E}e_2^2 = \cdots = \mathbb{E}e_p^2. \tag{8.24}$$

Further, we assume that data at two nodes i and j have less correlation when the distance between the nodes is large. So covariance matrix

$$R_e = \mathbb{V}e = \mathbb{E}(ee^\top) = (r_{ij})$$

can be given by

$$r_{ij} = \rho(d_{ij}) \tag{8.25}$$

for some decreasing function ρ and geodesic distance d_{ij} between nodes i and j. Note $r_{jj} = \rho(0)$ with the understanding that $d_{jj} = 0$ for all j. The off-diagonal entries of R_e are smaller than the diagonals.

Noise e can be further modeled as Gaussian white noise, i.e., Brownian motion or the generalized derivatives of the Wiener process, whose covariance matrix elements are Dirac-delta. For the discrete counterpart, $r_{ij} = \delta_{ij}$, where δ_{ij} is Kroneker-delta with $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. Thus,

$$R_e = \mathbb{E}(ee^\top) = I_p,$$

the identity matrix of size $p \times p$. Since $\delta_{jj} \geq \delta_{ij}$, Gaussian white noise is a special case of (8.25).

Once heat kernel smoothing is applied to (8.23), we have

$$K_\sigma * f = K_\sigma * \mu + K_\sigma * e. \tag{8.26}$$

We are interested in knowing how the statistical properties of the model change from (8.23) to (8.26). For $R_e = I_p$, the covariance matrix of smoothed noise is simply given as

$$R_{K_\sigma * e} = K_\sigma \mathbb{E}(ee^\top) K_\sigma = K_\sigma^2 = K_{2\sigma}.$$

We used the scale-space property of the heat kernel. In general, the covariance matrix of smoothed data $K_\sigma * e$ is given by

$$R_{K_\sigma * e} = K_\sigma \mathbb{E}(ee^\top) K_\sigma = K_\sigma R_e K_\sigma.$$

The variance of data will be often reduced after heat kernel smoothing in the following sense (Chung et al., 2005a,b):

Theorem 8.6 *Heat kernel smoothing reduces variability, i.e.,*

$$\mathbb{V}(K_\sigma * f)_j \le \mathbb{V} f_j$$

for all j. The subscript $_j$ indicates the jth element of the vector.

Proof. Note

$$\mathbb{V}(K_\sigma * f)_j = \mathbb{V}(K_\sigma * e)_j = \mathbb{E}\Big(\sum_{i=1}^{p} k_{ij} e_i\Big)^2.$$

Since (k_{ij}) is doubly stochastic, after applying Jensen's inequality, we obtain

$$\mathbb{E}\Big(\sum_{i=1}^{p} k_{ij} e_i\Big)^2 \le \mathbb{E}\Big(\sum_{i=1}^{p} k_{ij} e_i^2\Big) = \mathbb{E} e_i^2.$$

For the last equality, we used the equality of noise variability (8.24). Since $\mathbb{E} f_j = \mathbb{E} e_i^2$, we proved the statement. □

Theorem 8.6 shows that the variability of data decreases after heat kernel smoothing.

8.6.4 Skeleton Representation Using Heat Kernel Smoothing

Discrete heat kernel smoothing can be used to smooth out and present very complex patterns and get the skeleton representation. Here, we show how it is applied to the 3D graph obtained from the computed tomography (CT) of human lung vessel trees (Castillo et al., 2009; Wu et al., 2013; Chung et al., 2018a). In this example, the 3D binary vessel segmentation from CT was obtained using the multiscale Hessian filters at each voxel (Frangi et al., 1998; Korfiatis et al., 2011; Shang et al., 2011). The binary segmentation was converted into a 3D graph by taking each voxel as a node and connecting neighboring voxels. Using the 18-connected neighbor scheme, we connect two voxels only if they touch each other on their faces or edges. If voxels are only touching at their corner vertices, they are not considered as connected. If the six-connected neighbor scheme is used, we will obtain a far sparse adjacency matrix and corresponding graph Laplacian. The eigenvector of the graph Laplacian is obtained using an Implicitly Restarted Arnoldi Iteration Method (Lehoucq and Sorensen, 1996). We used 6,000 eigenvectors. Note we cannot have more eigenvectors than the number of nodes.

As an illustration, we performed heat kernel smoothing on simulated data. Gaussian noise is added to one of the coordinates (Figure 8.7). Heat kernel smoothing is performed on the noise added coordinate. Numbers in Figure 8.7 are kernel bandwidths. At $\sigma = 0$, heat kernel smoothing is equivalent to

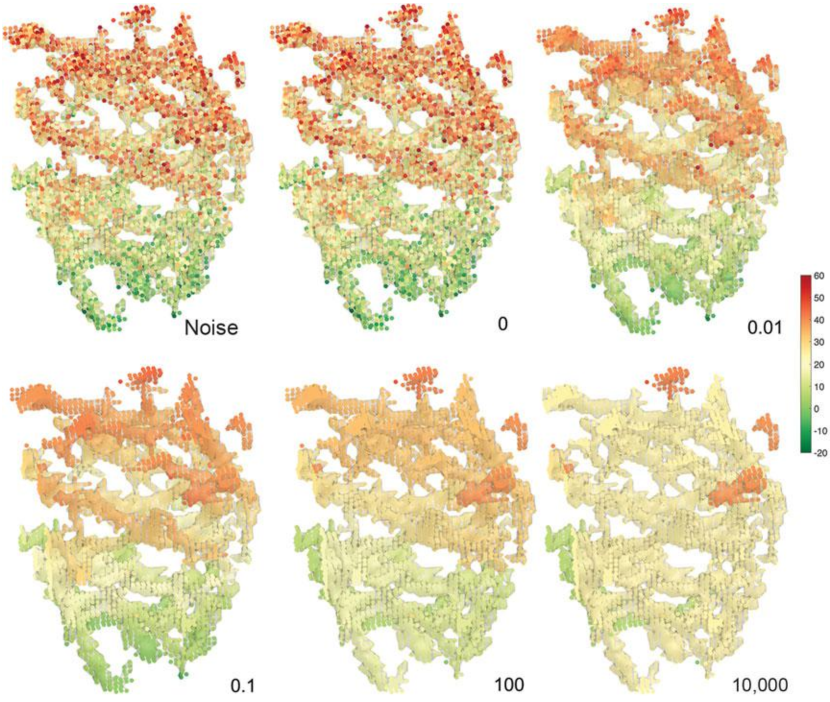


Figure 8.7 From top-left to right: 3D lung vessel tree. Gaussian noise is added to one of the coordinates. 3D graph constructed using six-connected neighbors. The numbers are the kernel bandwidth σ.

Fourier series expansion. Thus, we get the almost identical result. As the bandwidth increases, smoothing converges to the mean value. Each disconnected region should converge to its own different mean values. Thus, when $\sigma = 10{,}000$, the regions that are different colors are regions that are disconnected. This phenomenon is related to the hot spots conjecture in differential geometry (Banuelos and Burdzy, 1999; Chung et al., 2011b). The number of disconnected structures can be obtained counting the zero eigenvalues.

The technique can be used to extract the skeleton representation of vessel trees. We perform heat kernel smoothing on node coordinates with $\sigma = 1$. Then we rounded off the smoothed coordinates to the nearest integers. The rounded-off coordinates were used to reconstruct the binary segmentation. This gives the thick trees in Figure 8.8 (top-left). To obtain thinner trees, the smoothed coordinates were scaled by the factor of two, four, and six times before rounding off. This had the effect of increasing the image size relative to the kernel bandwidth, thus obtaining the skeleton representation of the complex blood vessel (Figure 8.8 clockwise from the top-right) (Lindvere et al., 2013; Cheng et al., 2014). By connecting the voxels sequentially, we can obtain the graph representation of the skeleton as well. The method can be easily adopted for obtaining the skeleton representation of complex brain network patterns.

8.6.5 Diffusion Wavelets

Consider a traditional wavelet basis $W_{t,q}(p)$ obtained from a mother wavelet W with scale and translation parameters t and q in Euclidean space (Kim et al., 2012b):

$$W_{t,q}(p) = \frac{1}{t} W\left(\frac{p-q}{t}\right). \tag{8.27}$$

The wavelet transform of a signal $f(p)$ is given by kernel

$$\langle W_{t,q}, f\rangle = \int_{\mathcal{M}} W_{t,q}(p) f(p)\, d\mu(p).$$

Scaling a function on an arbitrary manifold including graph is trivial. But the difficulty arises when one tries to translate a mother wavelet. It is not straightforward to generalize the Euclidean formulation (8.27) to an arbitrary manifold, due to the lack of regular grids (Nain et al., 2007; Bernal-Rusiel et al., 2008). The recent work based on the diffusion wavelet bypasses this problem also by taking bivariate kernel as a mother wavelet (Mahadevan and

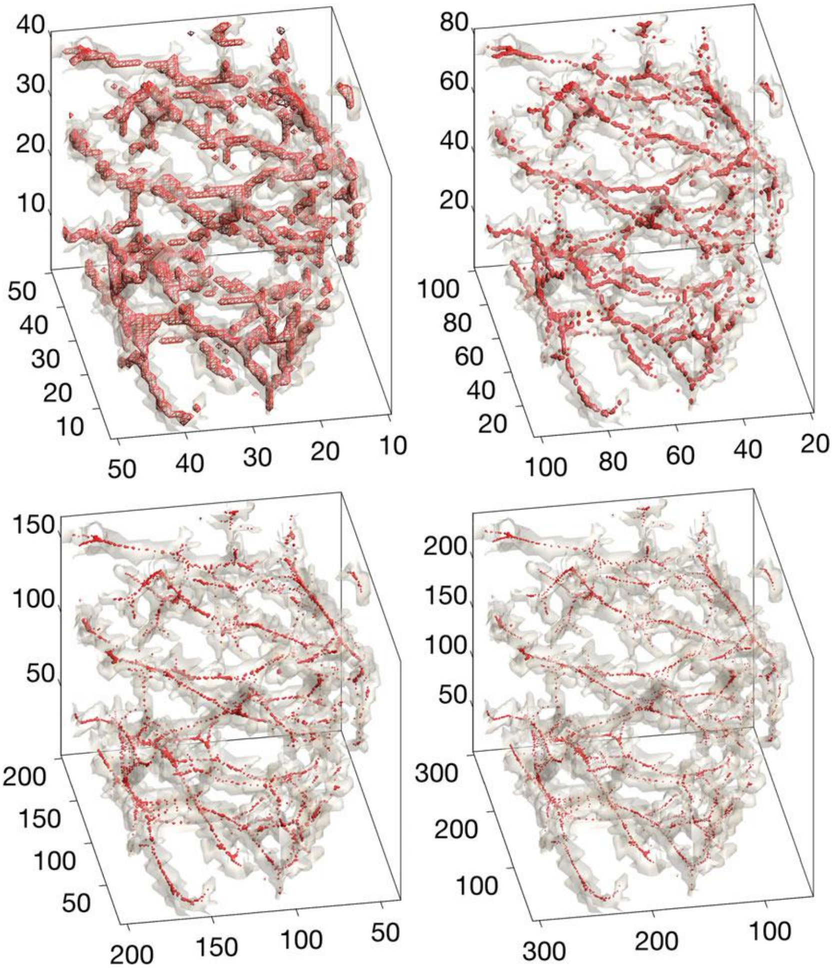


Figure 8.8 The skeleton representation of vessel trees. Using the heat kernel series expansion with bandwidth $\sigma = 1$ and 6,000 bases, we upsampled the binary segmentation at two, four, and six times (clockwise from top-right) larger than the original size (top-left).

Maggioni, 2006; Antoine et al., 2010; Hammond et al., 2011; Kim et al., 2012b). By simply changing the second argument of the kernel, it has the effect of translating the kernel. The diffusion wavelet construction has been fairly involving so far. However, it can be shown to be a special case of the heat kernel regression with proper normalization. Following the notations in

Antoine et al. (2010), Hammond et al. (2011), and Kim et al. (2012b), diffusion wavelet $W_{t,q}(p)$ at position p and scale t is given by

$$W_{t,q}(p) = \sum_{j=0}^{k} g(\lambda_j t)\psi_j(p)\psi_j(q),$$

for some scale function g. If we let $\tau_j = g(\lambda_j t)$, the diffusion wavelet transform is given by

$$\begin{aligned}\langle W_{t,q}, f\rangle &= \int_{\mathcal{M}} W_{t,q}(p) f(p)\, d\mu(p)\\ &= \sum_{j=0}^{k} g(\lambda_j t)\psi_j(q) \int_{\mathcal{M}} f(p)\psi_j(p)\, d\mu(p)\\ &= \sum_{j=0}^{k} \tau_j f_j \psi_j(q), \end{aligned} \tag{8.28}$$

where $f_j = \langle f, \psi_j\rangle$ is the Fourier coefficient. Note (8.28) is the kernel regression (Chung et al., 2015b). Hence, the diffusion wavelet transform can be simply obtained by doing the kernel regression without an additional wavelet machinery as done in Kim et al. (2012b). Further, if we let $g(\lambda_j t) = e^{-\lambda_j t}$, we have

$$W_{t,p}(q) = \sum_{j=0}^{k} e^{-\lambda_j t}\psi_j(p)\psi_j(q),$$

which is a heat kernel. The bandwidth t of heat kernel controls resolution while the translation is done by shifting one argument in the kernel.

8.7 Laplace Equation

In this section, we will show how to solve for steady state of diffusion on a graph. The distribution of fictional charges within the two boundaries sets up a scalar potential field Ψ, which satisfies the Poisson equation

$$\Delta\Psi = \frac{\partial^2\Psi}{\partial x^2} + \frac{\partial^2\Psi}{\partial y^2} + \frac{\partial^2\Psi}{\partial z^2} = \frac{\rho}{\epsilon_0},$$

where ρ is the total charge within the boundaries. If we set up the two boundaries at different potential, say at Ψ_0 and Ψ_1, without enclosing any charge, we have the Laplace equation

$$\Delta\Psi = 0.$$

By solving the Laplace equation with the two boundary condition, we obtain the potential field Ψ. Then the electric field perpendicular to the isopotential surfaces is given by $-\nabla\Psi$. The Laplace equation is mainly solved using the finite difference scheme. The electric field lines radiate from one conducting surface to the other without crossing each other. By tracing the electric field line on the graph, we obtain the geometric pattern of the graph.

The underlying framework is identical to the Laplace equation based surface flattening or cortical thickness estimation (Jones et al., 2000; Chung et al., 2010b). Without using the finite difference scheme, we can use an analytic approach for solving the Laplace equation in an arbitrary graph. The proposed method is essentially Galerkin's method (Kirby, 2000). Galerkin's method usually discretizes partial differential equations and integral equations as a collection of linear equations involving basis functions. The linear equations are then usually solved in the least squares fashion. The iterative residual fitting (IRF) algorithm (Chung et al., 2008a) can be considered as a special case of Galerkin's method.

The solution of the Laplace equation is approximated as a finite expansion

$$f(p) = \sum_{j=0}^{k} c_j \psi_j(p).$$

Consider following boundary conditions

$$f(p) = 1,\ p \in G_+, \text{ and } f(p) = -1,\ p \in G_-, \tag{8.29}$$

where G_+ and G_- are subgraphs of G. We may take G_+ and G_- at the two extreme nodes in the minimum spanning tree. The boundary conditions satisfy

$$1 = \sum_{j=0}^{k} c_j \psi_j(p_{2i}),\ p_{2i} \in G_+ \tag{8.30}$$

and

$$-1 = \sum_{j=0}^{k} c_j \psi_j(p_{3i}),\ p_{3i} \in G_-. \tag{8.31}$$

In the interior region $G\backslash(G_+ \cup G_-)$, by taking the Laplacian on the expansion $f(p) = \sum_{j=0}^{k} c_j \psi_j(p)$, we have

$$0 = \sum_{j=0}^{k} c_j \lambda_j \psi_j(p_{1i}),\ p_{1i} \in G\backslash(G_+ \cup G_-). \tag{8.32}$$

We may assume that the number of nodes in G_+ and G_- are substantially smaller than the number of nodes in $G\backslash(G_+ \cup G_-)$. So possibly we need to subsample the interior region. We assume that there are a, b, and c number of sampling nodes for equations (8.30), (8.31), and (8.32) respectively. We now combine linear equations (8.30), (8.31), and (8.32) together in a matrix form:

$$\underbrace{\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ -1 \end{pmatrix}}_{\mathbf{y}} = \underbrace{\begin{pmatrix} \lambda_1\psi_1(p_{11}) & \cdots & \lambda_k\psi_k(p_{11}) \\ \vdots & \ddots & \vdots \\ \lambda_1\psi_1(p_{1a}) & \cdots & \lambda_k(p_{1a}) \\ \psi_1(p_{21}) & \cdots & \psi_k(p_{21}) \\ \vdots & \ddots & \vdots \\ \psi_1(p_{2b}) & \cdots & \psi_k(p_{2b}) \\ \psi_1(p_{31}) & \cdots & \psi_k(p_{31}) \\ \vdots & \ddots & \vdots \\ \psi_1(p_{3c}) & \cdots & \psi_k(p_{3c}) \end{pmatrix}}_{\Psi} \underbrace{\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{k-1} \\ c_k \end{pmatrix}}_{\mathbf{C}}. \tag{8.33}$$

The preceding matrix equation can be solved by the least squares method:

$$\widehat{\mathbf{C}} = (\Psi'\Psi)^{-1}\Psi\mathbf{y}.$$

In order for the matrix $\Psi'\Psi$ to have the inverse, the total number of sampling voxels $(a + b + c)$ should be larger than the total number of basis k, which is likely to be true for brain images, so there is no need to use the pseudoinverse here.

9

Sparse Networks

There have been many attempts to identify high-dimensional network features via multivariate approaches (Worsley et al., 2005b; Lerch et al., 2006; He et al., 2007, 2008; Chung et al., 2013). Specifically, when the number of voxels or nodes, denoted as p, is substantially larger than the number of images, denoted as n, it produces an underdetermined model with infinitely many possible solutions. The *small-n large-p problem* is often remedied by regularizing the underdetermined system with additional sparse penalties.

Popular sparse network models include sparse correlations (Lee et al., 2011c; Chung et al., 2013, 2015a, 2017a), LASSO (Bickel and Levina, 2008; Huang et al., 2009; Peng et al., 2009; Chung et al., 2013), sparse canonical correlations (Avants et al., 2010), and graphical-LASSO (Banerjee et al., 2006, 2008; Friedman et al., 2008; Huang et al., 2009, 2010; Witten et al., 2011; Mazumder and Hastie, 2012). These popular sparse models require optimizing $L1$-norm penalties, which has been the major computational bottleneck for solving large-scale problems. Thus, many existing sparse brain network models in brain imaging have been restricted to a few hundreds nodes or less. The 2,527 MRI features used in a LASSO model for Alzheimer's disease (Xin et al., 2015) is probably the largest number of features used in any sparse model in the brain imaging literature.

9.1 Why Sparse Models?

If we are interested quantifying the measurements in every voxel in an image simultaneously, the standard procedure is to set up a multivariate general linear model (MGLM), which generalizes widely used univariate GLM by incorporating vector-valued responses and explanatory variables (Anderson, 1984; Friston et al., 1995; Worsley et al., 1996b, 2004; Taylor and Worsley, 2008). Hotelling's T^2-statistic is a special case of MGLM and has been mainly used for inference on surface shapes and deformations (Thompson et al., 1997; Joshi, 1998; Cao and Worsley, 1999a; Gaser et al., 1999; Chung et al., 2001b).

Let $\mathbf{J}_{n\times p} = (J_{ij})$ be the measurement matrix; J_{ij} is the measurement for subject i at voxel position j. The subscripts denote the dimension of matrix. We can think J_{ij} as either Jacobian determinant, fractional anisotropy values, or fMRI activation. Assume there are a total of n subjects and p voxels of interest. The measurement vector at the jth voxel is denoted as $\mathbf{x}_j = (J_{1j}, \cdots, J_{nj})^\top$. The measurement vector for the ith subject is denoted as $\mathbf{y}_i = (J_{i1}, \cdots, J_{ip})$, which is expected to be distributed identically and independently over subjects. Note that

$$\mathbf{J} = (\mathbf{x}_1, \cdots, \mathbf{x}_p) = (\mathbf{y}_1^\top, \cdots, \mathbf{y}_n^\top)^\top.$$

We may assume the covariance matrix of $\mathbf{y}_i$ to be

$$\mathbb{V}(\mathbf{y}_1) = \cdots = \mathbb{V}(\mathbf{y}_n) = \Sigma_{p\times p} = (\sigma_{kl}).$$

With these notations, we set up the following MGLM over all subjects and across different voxel positions:

$$\mathbf{J}_{n\times p} = \mathbf{X}_{n\times k}\mathbf{B}_{k\times p} + \mathbf{Z}_{n\times q}\mathbf{G}_{q\times p} + \mathbf{U}_{n\times p}\Sigma^{1/2}_{p\times p}, \tag{9.1}$$

where $\mathbf{X}$ is the matrix of contrasted explanatory variables while $\mathbf{B}$ is the matrix of unknown coefficients to be estimated. Nuisance covariates of noninterest are in the matrix $\mathbf{Z}$ and the corresponding coefficients are in the matrix $\mathbf{G}$. The components of Gaussian random matrix $\mathbf{U}$ are independently distributed with zero mean and unit variance. The symmetric matrix $\Sigma^{\mathbf{1/2}}$ is the square root of the covariance matrix accounting for the spatial dependency across different voxels. In MGLM (9.1), we are interested in testing the null hypothesis

$$H_0 : \mathbf{B} = 0.$$

The parameter matrices in the model are estimated via the least squares method. The resulting multivariate test statistics are called the Lawley–Hotelling trace or Roy's maximum root. When there is only one voxel, i.e., $p = 1$, these multivariate test statistics collapse to Hotelling's T^2-statistic (Worsley et al., 2004; Chung et al., 2010b).

Note that MGLM (9.1) is equivalent to the assumption that $\mathbf{y}_i$ follows multivariate normal with some mean μ and covariance Σ, i.e., $\mathbf{y}_i \sim N(\mu, \Sigma)$. Then neglecting constant terms, the log-likelihood function L of $\mathbf{y}_i$ is given by

$$L(\mu, \Sigma) = \log\det\Sigma^{-1} - \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \mu)^\top\Sigma^{-1}(\mathbf{y}_i - \mu).$$

By maximizing the log-likelihood, the MLE of μ and Σ are given by

$$\widehat{\mu} = \bar{\mathbf{y}}_i = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i$$

$$\widehat{\Sigma} = \frac{1}{n}\sum_{i=}^{n} (\mathbf{y}_i - \bar{\mathbf{y}}_i)^\top (\mathbf{y}_i - \bar{\mathbf{y}}_i). \tag{9.2}$$

For a notational convenience, we can center the measurement $\mathbf{y}_i$ such that

$$\mathbf{y}_i \leftarrow \mathbf{y}_i - \bar{\mathbf{y}}_i.$$

We are basically centering the measurements by subtracting the group mean over subjects. Then MLE (9.3) can be written in a more compact form

$$\widehat{\Sigma} = \frac{1}{n}\mathbf{J}_{p\times n}^\top \mathbf{J}_{n\times p}. \tag{9.3}$$

However, there is a serious defect with MGLM (9.1) and its MLE (9.3); namely the estimated covariance matrix $\widehat{\Sigma}$ is positive definite only for $n \geq p$ (Friston et al., 1995; Schäfer and Strimmer, 2005). $\mathbf{J}^\top\mathbf{J}$ becomes rank deficient for $n < p$. In most imaging studies, there are more voxels than the number of subjects, i.e., $n < p$. When $\widehat{\Sigma}$ is singular, we do not properly have the inverse of $\widehat{\Sigma}$, which is the precision matrix often needed in partial correlation-based network analyses (Lee et al., 2011c). This is the main reason MGLM was rarely employed over the whole brain region and researchers are still using mostly univariate approaches in imaging studies.

9.1.1 Why Sparse Networks?

The majority of functional and structural connectivity studies in brain imaging are usually performed following the standard analysis framework (Hagmann et al., 2007; Gong et al., 2009; Fornito et al., 2010; Zalesky et al., 2010). From 3D whole brain images, n regions of interest (ROI) are identified and serve as the nodes of the brain network. Measurements at ROIs are then correlated in a pairwise fashion to produce the connectivity matrix of size $n \times n$. The connectivity matrix is then thresholded to produce the adjacency matrix consisting of zeros and ones that define the link between two nodes. The binarized adjacency matrix is then used to construct the brain network. Then various graph complexity measures such as degree, clustering coefficients, entropy, path length, hub centrality, and modularity are defined on the graph, and the subsequent statistical inference is performed on these complexity measures.

For a large number of nodes, simple thresholding of correlation will produce a large number of edges, which makes the interpretation difficult. For example, for 3×10^5 voxels in an image, we can possibly have a total of 9×10^{10} directed edges in the graph. For this reason, we used the sparse data recovery framework in obtaining a far smaller number of significant edges.

9.2 Sparse Likelihood

Beyond sparse regression, others have proposed the likelihood methods. To remedy the small-n and large-p problem, the likelihood is regularized with a L1-norm penalty. If we center the measurements $\mathbf{y}_i$, the log-likelihood can be written as

$$\begin{aligned} L(\Sigma) &= \log\det\Sigma^{-1} - \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i^\top\Sigma^{-1}\mathbf{y}_i \\ &= \log\det\Sigma^{-1} - \operatorname{tr}\big(\Sigma^{-1}S\big), \end{aligned}$$

where $S = \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i^\top\mathbf{y}_i$ is the sample covariance matrix. We used the fact that the trace of a scalar value is equivalent to the scalar value itself and $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ for matrices A and B.

To avoid the small-n, large-p problem, we penalize the log-likelihood with L1-norm penalty:

$$L(\Sigma^{-1}) = \log\det\Sigma^{-1} - \operatorname{tr}\Big(\Sigma^{-1}S\Big) - \lambda\|\Sigma^{-1}\|_1, \tag{9.4}$$

where $\|\cdot\|_1$ is the sum of the absolute values of the elements. We made the likelihood as a function of Σ^{-1} to simply emphasize that we are trying to estimate the inverse covariance matrix. The penalized log-likelihood is maximized over the space of all possible symmetric positive definite matrices. (9.4) is a convex problem and it is usually solved using the graphical-LASSO (GLASSO) algorithm (Banerjee et al., 2006, 2008; Friedman et al., 2008; Huang et al., 2010; Mazumder and Hastie, 2012). The tuning parameter $\lambda > 0$ controls the sparsity of the off-diagonal elements of the inverse covariance matrix. By increasing $\lambda > 0$, the estimated inverse covariance matrix becomes more sparse.

GLASSO is a fairly time-consuming algorithm (Friedman et al., 2008; Huang et al., 2010). Solving GLASSO for 548 nodes, for instance, may take up to 6 minutes on slow desktop computers if fast algorithms like Hsieh et al. (2013) is not used. If $\Sigma_i^{-1}(\lambda)$ is the estimated inverse sparse covariance for group i at given sparse parameter λ, we are interested in testing the equivalence of inverse covariance matrices between the two groups at fixed λ, i.e.,

$$H_0 : \Sigma_1^{-1}(\lambda) = \Sigma_2^{-1}(\lambda).$$

9.2.1 Filtration in Graphical-LASSO

The solution to graphical-LASSO has a peculiar nested topological structure. Let $\Sigma^{-1}(\lambda) = (\sigma^{ij}(\lambda))$ be the inverse covariance estimated from graphical-LASSO. Let $A(\lambda) = (a_{ij})$ be the corresponding adjacency matrix given by

$$a_{ij}(\lambda) = \begin{cases} 1 & \text{if } \widehat{\sigma}^{ij} \neq 0; \\ 0 & \text{otherwise.} \end{cases} \tag{9.5}$$

The adjacency matrix A induces a graph $\mathcal{G}(\lambda)$ consisting of $\kappa(\lambda)$ number of partitioned subgraphs

$$\mathcal{G}(\lambda) = \bigcup_{l=1}^{\kappa(\lambda)} G_l(\lambda) \ \text{ with } \ G_l = \{V_l(\lambda), A_l(\lambda)\},$$

where V_l and A_l are node and edge sets of subgraph G_l.

Let $S = (s_{ij})$ be the sample covariance matrix. Let $B(\lambda) = (b_{ij})$ be the adjacency matrix defined by

$$b_{ij}(\lambda) = \begin{cases} 1 & \text{if } |\widehat{s}_{ij}| > \lambda; \\ 0 & \text{otherwise.} \end{cases} \tag{9.6}$$

The adjacency matrix B similarly induces a graph with $\tau(\lambda)$ disjoint subgraphs:

$$\mathcal{H}(\lambda) = \bigcup_{l=1}^{\tau(\lambda)} H_l(\lambda) \ \text{ with } H_l = \{W_l(\lambda), B_l(\lambda)\},$$

where W_l and B_l are node and edge sets of subgraph H_l. Then the partitioned graphs are shown to be partially nested in a sense that the node sets exhibits persistency.

Theorem 9.1 *For any* $\lambda > 0$*, the adjacency matrices* (9.5) *and* (9.6) *induce the identical vertex partition so that* $\kappa(\lambda) = \tau(\lambda)$ *and* $V_l(\lambda) = W_l(\lambda)$*. Further, the node sets* V_l *and* W_l *form filtrations over the sparse parameter:*

$$V_l(\lambda_1) \supset V_l(\lambda_2) \supset V_l(\lambda_3) \supset \cdots \tag{9.7}$$

$$W_l(\lambda_1) \supset W_l(\lambda_2) \supset W_l(\lambda_3) \supset \cdots \tag{9.8}$$

for $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \cdots$.

From (9.6), it is trivial to see the filtration holds for W_l. The filtration for V_l is proved in Huang et al. (2010). The equivalence of the node sets $V_l = W_l$ is proved in Mazumder and Hastie (2012). Note that the edge sets may not form a filtration (Figure 9.1). The construction of the filtration on the node sets V_l

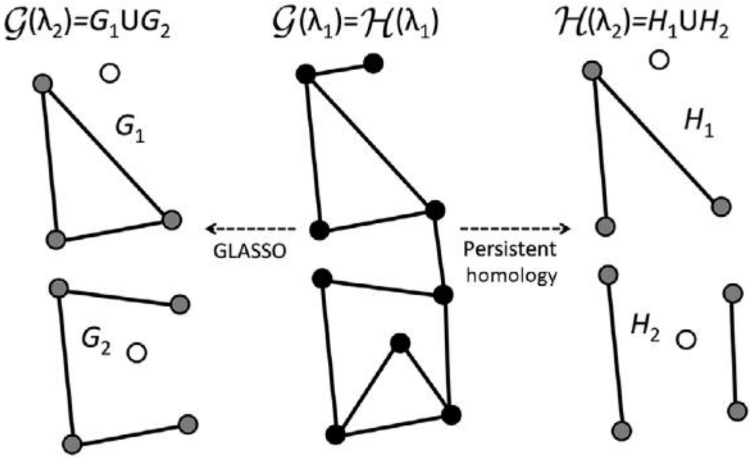

$\mathcal{G}(\lambda_2)=G_1 \cup G_2$
$\mathcal{G}(\lambda_1)=\mathcal{H}(\lambda_1)$
$\mathcal{H}(\lambda_2)=H_1 \cup H_2$
G_1
G_2
GLASSO
Persistent homology
H_1
H_2

Figure 9.1 Schematic of graph filtrations obtained by sparse-likelihood (9.5) and sample covariance thresholding (9.6). The vertex set of $\mathcal{G}(\lambda_1) = \mathcal{H}(\lambda_1)$ consists of black nodes. For the next filtration value λ_2, $\mathcal{G}(\lambda_2) \neq \mathcal{H}(\lambda_2)$ since the edge sets are different. However, the partitioned vertex sets (gray colored) of $\mathcal{G}(\lambda_2)$ and $\mathcal{H}(\lambda_2)$ match.

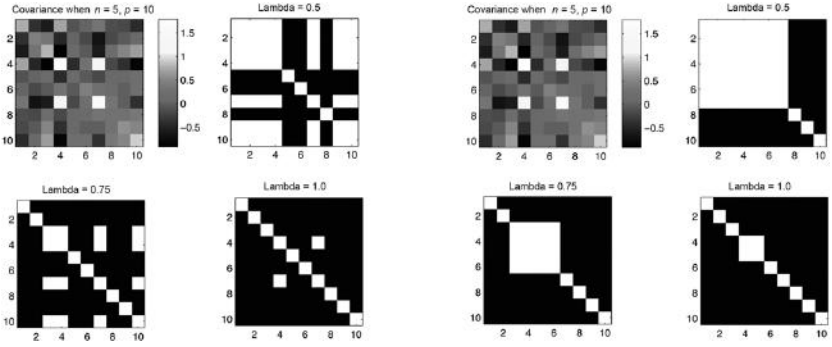


Figure 9.2 Left: adjacency matrices obtained through graphical-LASSO with increasing λ values. The persistent homological structure is self-evident. Right: adjacency matrices are clustered as a block diagonal matrix D by permutation.

(9.7) is very time consuming since we have to solve the sequence of graphical-LASSO. For instance, for 548 node sets and 547 different filtration values, the whole filtration takes more than 54 hours in a desktop (Chung et al., 2015a).

In Figure 9.2, we randomly simulated the data matrix $X_{5\times 10}$ from the standard normal distribution. The sample covariance matrix is then fed into graphical-LASSO with different filtration values. To identify the structure better, we transformed the adjacency matrix A by permutation P such that

$D = PAP^{-1}$ is a block diagonal matrix. Theoretically only the partitioned node sets are expected to exhibit the nestedness, but in this example the edge sets are also nested as well.

9.3 Sparse Correlation Network

The problem with graphical-LASSO or any type of similar L1 norm optimization is that it becomes computationally expensive as the number of node p increases. So it is not really practical for large-scale brain networks. In this section, we present a scalable large-scale network model ($p > 25{,}000$) that yields greater computational speed and efficiency by bypassing the computational bottleneck of optimizing $L1$-penalties.

There are few previous studies at speeding up the computation for sparse models. By identifying block diagonal structures in the estimated (inverse) covariance matrix, it is possible to reduce the computational burden in the penalized log-likelihood method (Witten et al., 2011; Mazumder and Hastie, 2012). However, the method presented in this section differs from Mazumder and Hastie (2012) and Witten et al. (2011) in that we do not need to assume that the data will follow Gaussianness. Subsequently, there is no need to specify the likelihood function. Further, the cost functions we are optimizing are different. Specifically, we propose a novel sparse network model based on *correlations*. Although correlations are often used in sciences in connection to times series and stochastic processes (Worsley et al., 2005a,b), the sparse version of correlation has been somewhat neglected.

Consider measurement vector $\mathbf{x}_j$ on node j. If we center and rescale the measurement $\mathbf{x}_j$ such that

$$\| \mathbf{x}_j \|^2 = \mathbf{x}_j^\top \mathbf{x}_j = 1,$$

the sample correlation between nodes i and j is given by $\mathbf{x}_i^\top \mathbf{x}_j$. Since the data are normalized, the sample covariance matrix is reduced to the sample correlation matrix.

Consider the following linear regression between nodes j and k ($k \neq j$):

$$\mathbf{x}_j = \gamma_{jk} \mathbf{x}_k + \epsilon_j. \tag{9.9}$$

We are basically correlating data at node j to data at node k. In this particular case, γ_{jk} is the usual Pearson correlation. The least squares estimation (LSE) of γ_{jk} is then given by

$$\widehat{\gamma}_{jk} = \mathbf{x}_j^\top \mathbf{x}_k, \tag{9.10}$$

which is the sample correlation. For the normalized data, regression coefficient estimation is exactly the sample correlation. For the normalized and centered data, the regression coefficient is the correlation. It can be shown that (9.22) minimizes the sum of least squares over all nodes:

$$\sum_{j=1}^{p} \sum_{k \neq j} \parallel \mathbf{x}_j - \gamma_{jk}\mathbf{x}_k \parallel^2 . \tag{9.11}$$

Note that we do not really care about correlating $\mathbf{x}_j$ to itself since the correlation is then trivially $\gamma_{jj} = 1$.

9.3.1 Sparse Correlations

Let $\Gamma = (\gamma_{jk})$ be the correlation matrix. The sparse penalized version of (9.11) is given by

$$F(\Gamma) = \frac{1}{2} \sum_{j=1}^{p} \sum_{k \neq j} \parallel \mathbf{x}_j - \gamma_{jk}\mathbf{x}_k \parallel^2 + \lambda \sum_{j=1}^{p} \sum_{k \neq j} |\gamma_{jk}|. \tag{9.12}$$

The sparse correlation is given by minimizing $F(\Gamma)$. By increasing λ, the estimated correlation matrix $\widehat{\Gamma}(\lambda)$ becomes more sparse. When $\lambda = 0$, the sparse correlation is simply given by the sample correlation, i.e., $\widehat{\gamma}_{jk} = \mathbf{x}_j^\top \mathbf{x}_k$. As λ increases, the correlation matrix Γ shrinks to zero and becomes more sparse. This is separable compressed sensing or LASSO type problem. However, there is no need to numerically optimize (9.12) using the coordinate descent learning or the active-set algorithm often used in compressed sensing (Friedman et al., 2008; Peng et al., 2009). The minimization of (9.12) can be done by the proposed soft-thresholding method analytically by exploiting the topological structure of the problem. Since $\mathbf{x}_i^\top \mathbf{x}_j \neq \delta_{ij}$, the Dirac delta, it looks like the sparse regression is not orthogonal design and the existing soft-thresholding method for LASSO (Tibshirani, 1996) is not directly applicable. However, it can be made into orthogonal design. The detail is given in the sparse cross-correlation section.

Theorem 9.2 *For $\lambda \geq 0$, the solution of the following separable LASSO problem*

$$\widehat{\gamma}_{jk}(\lambda) = \arg\min_{\gamma_{jk}} \frac{1}{2} \sum_{j=1}^{p} \sum_{k \neq j} \parallel \mathbf{x}_j - \gamma_{jk}\mathbf{x}_k \parallel^2 + \lambda \sum_{j=1}^{p} \sum_{k \neq j} |\gamma_{jk}|,$$

is given by the soft-thresholding

$$\widehat{\gamma}_{jk}(\lambda) = \begin{cases} \mathbf{x}_j^\top \mathbf{x}_k - \lambda & \textit{if}\, \mathbf{x}_j^\top \mathbf{x}_k > \lambda \\ 0 & \textit{if}\, |\mathbf{x}_j^\top \mathbf{x}_k| \le \lambda \\ \mathbf{x}_j^\top \mathbf{x}_k + \lambda & \textit{if}\, \mathbf{x}_j^\top \mathbf{x}_k < -\lambda \end{cases}. \tag{9.13}$$

Proof. Write (9.12) as

$$F(\Gamma) = \frac{1}{2} \sum_{j=1}^{p} \sum_{k \neq j} f(\gamma_{jk}), \tag{9.14}$$

where

$$f(\gamma_{jk}) = \parallel \mathbf{x}_j - \gamma_{jk}\mathbf{x}_k \parallel^2 + 2\lambda|\gamma_{jk}|.$$

Since $f(\gamma_{jk})$ is nonnegative and convex, $F(\Gamma)$ is minimum if each component $f(\gamma_{jk})$ achieves minimum. So we only need to minimize each component $f(\gamma_{jk})$. This differentiates our sparse correlation formulation from the standard compressed sensing that cannot be optimized in this componentwise fashion. $f(\gamma_{jk})$ can be rewritten as

$$\begin{aligned} f(\gamma_{jk}) &= \|\mathbf{x}_j\|^2 - 2\gamma_{jk}\mathbf{x}_j^\top \mathbf{x}_k + \gamma_{jk}^2 \|\mathbf{x}_k\|^2 + 2\lambda|\gamma_{jk}| \\ &= (\gamma_{jk} - \mathbf{x}_j^\top \mathbf{x}_k)^2 + 2\lambda|\gamma_{jk}| + 1. \end{aligned}$$

We used the fact $\mathbf{x}_j^\top \mathbf{x}_j = 1$.

For $\lambda = 0$, the minimum of $f(\gamma_{jk})$ is achieved when $\gamma_{jk} = \mathbf{x}_j^\top \mathbf{x}_k$, which is the usual LSE. For $\lambda > 0$, since $f(\gamma_{jk})$ is quadratic in γ_{jk}, the minimum is achieved when

$$\frac{\partial f}{\partial \gamma_{jk}} = 2\gamma_{jk} - 2\mathbf{x}_j^\top \mathbf{x}_k \pm 2\lambda = 0. \tag{9.15}$$

The sign of λ depends on the sign of γ_{jk}. Thus, sparse correlation $\widehat{\gamma}_{jk}$ is given by a soft-thresholding of $\mathbf{x}_j^\top \mathbf{x}_k$:

$$\widehat{\gamma}_{jk}(\lambda) = \begin{cases} \mathbf{x}_j^\top \mathbf{x}_k - \lambda & \text{if } \mathbf{x}_j^\top \mathbf{x}_k > \lambda \\ 0 & \text{if } |\mathbf{x}_j^\top \mathbf{x}_k| \le \lambda \\ \mathbf{x}_j^\top \mathbf{x}_k + \lambda & \text{if } \mathbf{x}_j^\top \mathbf{x}_k < -\lambda \end{cases}. \tag{9.16}$$

□

The estimated sparse correlation (9.16) basically thresholds the sample correlation that is larger or smaller than λ by the amount λ. Due to this simple expression, there is no need to optimize (9.12) numerically as is often done in compressed sensing or LASSO (Friedman et al., 2008; Peng et al., 2009). However, Theorem 9.2 is only applicable to separable cases and for nonseparable cases, and numerical optimization is still needed.

The different choices of sparsity parameter λ will produce different solutions in sparse model $\mathcal{A}(\lambda)$. Instead of analyzing each model separately, we can analyze the whole collection of all the sparse solutions for many different values of λ. This avoids the problem of identifying the optimal sparse parameter that may not be optimal in practice. The question is then how to use the collection of $\mathcal{A}(\lambda)$ in a coherent mathematical fashion. This can be addressed using persistent homology (Edelsbrunner and Harer, 2008; Lee et al., 2011a, 2012).

9.3.2 Filtration in Sparse Correlations

Using the sparse solution (9.16), we can construct a filtration. We will basically build a graph $\mathcal{G}$ using spare correlations. Let $\widehat{\gamma}_{jk}(\lambda)$ be the sparse correlation estimate. Let $A(\lambda) = (a_{ij})$ be the adjacency matrix defined as

$$a_{jk}(\lambda) = \begin{cases} 1 & \text{if } \widehat{\gamma}_{jk}(\lambda) \neq 0; \\ 0 & \text{otherwise.} \end{cases}$$

This is equivalent to the adjacency matrix $B = (b_{jk})$ defined as

$$b_{jk}(\lambda) = \begin{cases} 1 & \text{if } |\mathbf{x}_j^\top \mathbf{x}_k| > \lambda; \\ 0 & \text{otherwise.} \end{cases} \tag{9.17}$$

The adjacency matrix B is simply obtained by thresholding the sample correlations. Then the adjacency matrices A and B induce a identical graph $\mathcal{G}(\lambda)$ consisting of $\kappa(\lambda)$ number of partitioned subgraphs

$$\mathcal{G}(\lambda) = \bigcup_{l=1}^{\kappa(\lambda)} G_l(\lambda) \ \text{ with } G_l = \{V_l(\lambda), E_l(\lambda)\},$$

where V_l and E_l are node and edge sets respectively. Note

$$G_l \bigcap G_m = \varnothing \ \text{ for any } \ l \neq m$$

and no two nodes between the different partitions are connected. The node and edge sets are denoted as $\mathcal{V}(\lambda) = \bigcup_{l=1}^{\kappa} V_l$ and $\mathcal{E}(\lambda) = \bigcup_{l=1}^{\kappa} E_l$ respectively. Then we have the following theorem:

Theorem 9.3 *The induced graph from the spare correlation forms a filtration:*

$$\mathcal{G}(\lambda_1) \supset \mathcal{G}(\lambda_2) \supset \mathcal{G}(\lambda_3) \supset \cdots \tag{9.18}$$

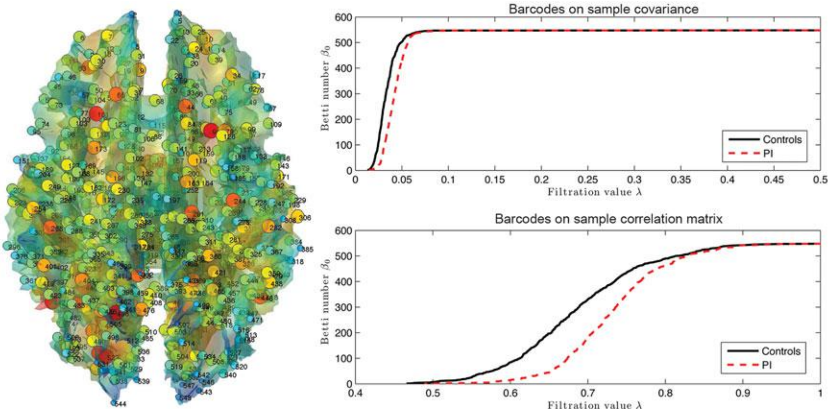


Figure 9.3 Jacobian determinants of the deformation field are measured at 548 nodes along the white matter boundary (Chung et al., 2015a). The β_0-number (number of connected components) of the filtrations on the sample correlations and covariances show huge group separation between normal controls and postinstitutionalized (PI) children.

for $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \cdots$. *Equivalently, the node and edge sets also form filtrations as well:*

$$\mathcal{V}(\lambda_1) \supset \mathcal{V}(\lambda_2) \supset \mathcal{V}(\lambda_3) \supset \cdots$$
$$\mathcal{E}(\lambda_1) \supset \mathcal{E}(\lambda_2) \supset \mathcal{E}(\lambda_3).$$

The proof can be easily obtained from the definition of adjacency matrix (9.17) (see Figure 9.3).

9.3.3 Sparse Cross-Correlations

We can extend the sparse correlation framework to the sparse cross-correlations. Let $V = \{v_1, \cdots, v_p\}$ be a node set where data are observed. We expect the number of nodes p to be significantly larger than the number of images n, i.e., $p \gg n$. Let $x_k(v_i)$ and $y_k(v_i)$ be the kth paired scalar measurements at node v_i. They can be twins, longitudinal scans, or even multimodal images. Denote $\mathbf{x}(v_i) = (x_1(v_i), \cdots, x_n(v_i))^\top$ and let $\mathbf{y}(v_i) = (y_1(v_i), \cdots, y_n(v_i))^\top$ be the paired data vectors over n different images at voxel v_i. Center and scale $\mathbf{x}$ and $\mathbf{y}$ such that

$$\sum_{k=1}^{n} x_k(v_i) = \sum_{k=1}^{n} y_k(v_i) = 0,$$

$$\|\mathbf{x}(v_i)\|^2 = \mathbf{x}^\top(v_i)\mathbf{x}(v_i) = \|\mathbf{y}(v_i)\|^2 = \mathbf{y}^\top(v_i)\mathbf{y}(v_i) = 1$$

for all v_i. The reasons for centering and scaling will soon be obvious.

We set up a hypernetwork by relating the paired vectors at different voxels v_i and v_j:

$$\mathbf{y}(v_j) = \sum_{i=1}^{p} \beta_{ij}\ \mathbf{x}(v_i) + \mathbf{e} \tag{9.19}$$

for some zero-mean noise vector $\mathbf{e}$ (Figure 9.4). The parameters $\beta = (\beta_{ij})$ are the weights of the hyperedges between voxels v_i and v_j that have to be estimated. We are constructing a physically nonexistent artificial network across different images. For fMRI, (9.19) requires estimating over billions of connections, which is computationally challenging. In practice, however, each application will likely to force β to have a specific structure that may reduce the computational burden.

For this section, let us set up a linear model between $\mathbf{x}(v_i)$ and $\mathbf{y}(v_j)$:

$$\mathbf{y}(v_j) = b_{ij}\ \mathbf{x}(v_i) + \mathbf{e}, \tag{9.20}$$

where $\mathbf{e}$ is the zero-mean error vector whose components are independent and identically distributed. Since the data are all centered, we do not have the

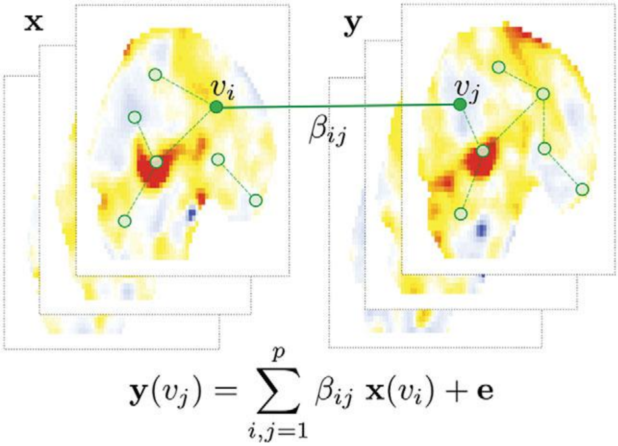

$\mathbf{x}$
$\mathbf{y}$
v_i
v_j
β_{ij}
$\mathbf{y}(v_j) = \sum_{i,j=1}^{p} \beta_{ij}\ \mathbf{x}(v_i) + \mathbf{e}$

Figure 9.4 The schematic of hypernetwork construction on paired image vectors $\mathbf{x}$ and $\mathbf{y}$. The image vectors $\mathbf{y}$ at voxel v_j are modeled as a linear combination of the first image vector $\mathbf{x}$ at all other voxels. The estimated parameters β_{ij} give the hyperedge weights.

intercept in linear regression (9.20). The LSE of b_{ij} that minimizes the L2-norm

$$\sum_{i,j=1}^{p} \parallel \mathbf{y}(v_j) - b_{ij}\ \mathbf{x}(v_i) \parallel^2 \tag{9.21}$$

is given by

$$\widehat{b_{ij}} = \mathbf{x}^\top(v_i)\mathbf{y}(v_j), \tag{9.22}$$

which are the (sample) *cross-correlations* (Worsley et al., 2005a,b). The cross-correlation is invariant under the centering and scaling operations. The sparse version of L2-norm (9.21) is given by

$$F(\beta; \mathbf{x}, \mathbf{y}, \lambda) = \frac{1}{2}\sum_{i,j=1}^{p} \parallel \mathbf{y}(v_j) - \beta_{ij}\ \mathbf{x}(v_i) \parallel^2 + \lambda \sum_{i,j=1}^{p} |\beta_{ij}|. \tag{9.23}$$

The *sparse cross-correlation* is then obtained by minimizing over every possible $\beta_{ij} \in \mathbb{R}$:

$$\widehat{\beta}(\lambda) = \arg\min_{\beta} F(\beta; \mathbf{x}, \mathbf{y}, \lambda). \tag{9.24}$$

The estimated sparse cross-correlations $\widehat{\beta}(\lambda) = (\widehat{\beta_{ij}}(\lambda))$ shrink toward zero as sparse parameter $\lambda \geq 0$ increases. The direct optimization of (9.23) for large p is computationally demanding. However, there is no need to optimize (9.23) numerically using the coordinate descent learning or the active-set algorithm as often done in sparse optimization (Friedman et al., 2008; Peng et al., 2009). We can show that the minimization of (9.23) is simply done algebraically.

Theorem 9.4 *For $\lambda \geq 0$, the minimizer of $F(\beta; \mathbf{x}, \mathbf{y}, \lambda)$ is given by*

$$\widehat{\beta_{ij}}(\lambda) = \begin{cases} \mathbf{x}^\top(v_i)\mathbf{y}(v_j) - \lambda & \textit{if } \mathbf{x}^\top(v_i)\mathbf{y}(v_j) > \lambda \\ 0 & \textit{if } |\mathbf{x}^\top(v_i)\mathbf{y}(v_j)| \leq \lambda \\ \mathbf{x}^\top(v_i)\mathbf{y}(v_j) + \lambda & \textit{if } \mathbf{x}^\top(v_i)\mathbf{y}(v_j) < -\lambda \end{cases}. \tag{9.25}$$

Although it is not obvious, Theorem 9.4 is related to the orthogonal design in LASSO (Tibshirani, 1996) and the soft-shrinkage in wavelets (Donoho et al., 1995). To see this, let us transform linear equations (9.20) into a index-free matrix equation:

$$\begin{bmatrix} \mathbf{y}(v_1) & \cdots & \mathbf{y}(v_1) \\ \mathbf{y}(v_2) & \cdots & \mathbf{y}(v_2) \\ \vdots & \ddots & \vdots \\ \mathbf{y}(v_p) & \cdots & \mathbf{y}(v_p) \end{bmatrix} = \begin{bmatrix} b_{11}\mathbf{x}(v_1) & b_{21}\mathbf{x}(v_2) & \cdots & b_{p1}\mathbf{x}(v_p) \\ b_{12}\mathbf{x}(v_1) & b_{22}\mathbf{x}(v_2) & \cdots & b_{p2}\mathbf{x}(v_p) \\ \vdots & \vdots & \ddots & \vdots \\ b_{1p}\mathbf{x}(v_1) & b_{2p}\mathbf{x}(v_2) & \cdots & b_{pp}\mathbf{x}(v_p) \end{bmatrix} + \begin{bmatrix} \mathbf{e} & \cdots & \mathbf{e} \\ \mathbf{e} & \cdots & \mathbf{e} \\ \vdots & \ddots & \vdots \\ \mathbf{e} & \cdots & \mathbf{e} \end{bmatrix}.$$

The preceding matrix equation can be vectorized as follows:

$$\left[\begin{array}{c} \mathbf{y}(v_1) \\ \vdots \\ \mathbf{y}(v_p) \\ \hline \vdots \\ \hline \mathbf{y}(v_1) \\ \vdots \\ \mathbf{y}(v_p) \end{array}\right] = \left[\begin{array}{cccccc} \mathbf{x}(v_1)\cdots & 0 & & & & \\ \vdots \;\; \ddots & \vdots & \cdots & & \mathbf{0} & \\ 0 & \cdots\mathbf{x}(v_1) & & & & \\ \vdots & & \ddots & & \vdots & \\ & & & \mathbf{x}(v_p)\cdots & 0 & \\ \mathbf{0} & & \cdots & \vdots \;\; \ddots & \vdots & \\ & & & 0 & \cdots\mathbf{x}(v_p) & \end{array}\right] \left[\begin{array}{c} b_{11} \\ \vdots \\ b_{p1} \\ \hline \vdots \\ \hline b_{1p} \\ \vdots \\ b_{pp} \end{array}\right] + \left[\begin{array}{c} \mathbf{e} \\ \vdots \\ \mathbf{e} \\ \hline \vdots \\ \hline \mathbf{e} \\ \vdots \\ \mathbf{e} \end{array}\right].$$

The preceding equation can be written in a more compact form. Let

$$\begin{aligned} \mathbf{X}_{n\times p} &= [\mathbf{x}(v_1)\ \mathbf{x}(v_2)\cdots\ \mathbf{x}(v_p)] \\ \mathbf{Y}_{n\times p} &= [\mathbf{y}(v_1)\ \mathbf{y}(v_2)\cdots\ \mathbf{y}(v_p)] \\ \mathbf{1}_{a\times b} &= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{a\times b}. \end{aligned}$$

Then the matrix equation can be written as

$$\mathbf{1}_{p\times 1} \otimes vec(\mathbf{Y}) = \mathbb{X}_{np^2\times p^2}\ vec(b) + \mathbf{1}_{np^2\times 1} \otimes \mathbf{e}, \tag{9.26}$$

where *vec* is the vectorization operation. The block diagonal design matrix $\mathbb{X}$ consists of p diagonal blocks $I_p \otimes \mathbf{x}(v_1), \cdots, I_p \otimes \mathbf{x}(v_p)$, where I_p is the $p \times p$ identity matrix. Subsequently, $\mathbb{X}^\top\mathbb{X}$ is again a block diagonal matrix, where the ith block is

$$[I_p \otimes \mathbf{x}(v_i)]^\top [I_p \otimes \mathbf{x}(v_i)] = I_p \otimes [\mathbf{x}(v_i)^\top \mathbf{x}(v_i)] = I_p.$$

Thus, $\mathbb{X}$ is an orthogonal design. However, our formulation is *not* exactly the orthogonal design of LASSO as specified in Tibshirani (1996) since the noise components in (9.26) are not independent. Further, in standard LASSO, there are more columns than rows in $\mathbb{X}$. In our case, there are n times more rows. Still the soft-thresholding method introduced in Tibshirani (1996) is applicable and we obtain the analytic solution, which speeds up the computation drastically compared to existing LASSO-based numerical optimization (Figure 9.5) (Friedman et al., 2008; Peng et al., 2009).

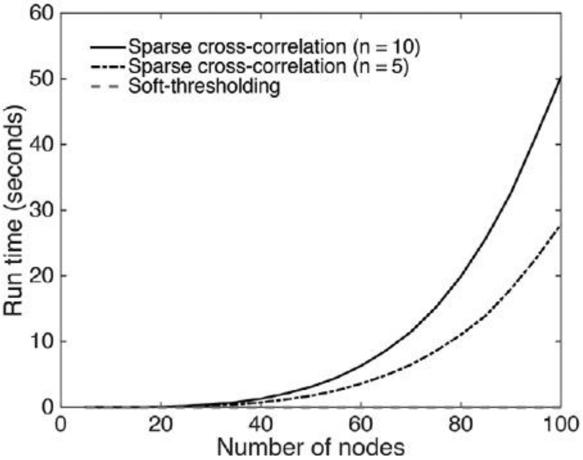


Figure 9.5 Run time comparison of estimating sparse cross-correlations. The LASSO-based numerical optimization with $n = 5, 10$ images with varying numbers of nodes. The run time scales linearly with the number of images but scales exponentially with the number of nodes. The LASSO runs more than 100,000 times slower compared to the soft-thresholding method for $p = 100$ nodes and $n = 10$ images.

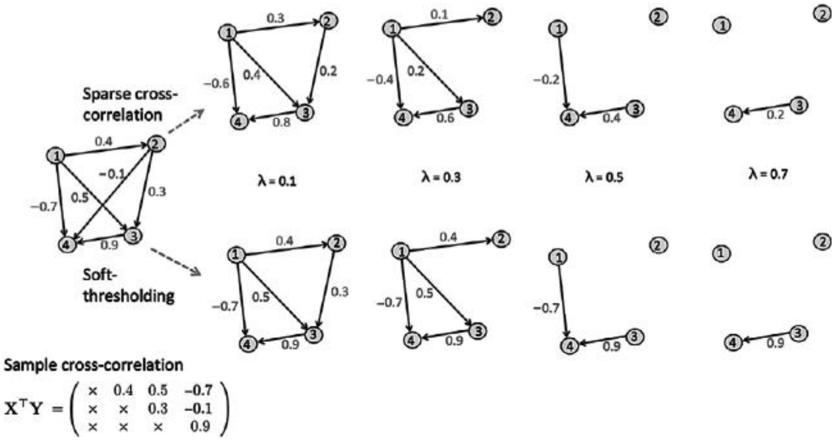


Figure 9.6 Schematic showing the equivalence of binary graph construction using the sparse cross-correlations and soft-thresholding. Top: the sparse cross-correlations are estimated by minimizing the L_1 cost function (9.23) for four different sparse parameters λ. The edge weights shrinked to zero are removed. Bottom: the equivalent binary graph can be obtained by soft-thresholding, i.e., simply thresholding the sample cross-correlations at λ.

Theorem 9.4 generalizes the sparse correlation case given in Chung et al. (2013). Figure 9.6-top displays an example of obtaining sparse cross-correlations from the initial sample cross-correlation matrix

$$\mathbf{X}^\top\mathbf{Y} = \begin{pmatrix} \times & 0.4 & 0.5 & -0.7 \\ \times & \times & 0.3 & -0.1 \\ \times & \times & \times & 0.9 \end{pmatrix}$$

using Theorem 9.4. Due to directional nature of the cross-correlation matrix, only the upper triangle part of the sample cross-correlation is demonstrated.

9.4 Partial Correlation Network

Let p be the number of nodes in the network. In most applications, the number of nodes is expected to be larger than the number of observations n, which gives an underdetermined system. Consider measurement vector at the jth node

$$\mathbf{x}_j = (x_{1j}, \cdots, x_{nj})^\top$$

consisting of n measurements. Vector $\mathbf{x}_j$ are assumed to be distributed with mean zero and covariance $\Sigma = (\sigma_{ij})$. The correlation γ_{ij} between the two nodes i and j is given by

$$\gamma_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}.$$

By thresholding the correlation, we can establish a link between two nodes. However, there is a problem with this simplistic approach in that it fails to explicitly factor out the confounding effect of other nodes. To remedy this problem, partial correlations can be used in factoring out the dependency of other nodes (Marrelec et al., 2006; He et al., 2007; Huang et al., 2009, 2010; Peng et al., 2009).

If we denote the inverse covariance matrix as $\Sigma^{-1} = (\sigma^{ij})$, the *partial correlation* between the nodes i and j while factoring out the effect of all other nodes is given by (Peng et al., 2009)

$$\rho_{ij} = -\frac{\sigma^{ij}}{\sqrt{\sigma^{ii}\sigma^{jj}}}. \tag{9.27}$$

Equivalently, we can compute the partial correlation via a linear model as follows. Consider a linear model of correlating measurement at node j to all other nodes:

$$\mathbf{x}_j = \sum_{k \neq j} \beta_{jk} \mathbf{x}_k + \epsilon_k. \tag{9.28}$$

The parameters β_{jk} are estimated by minimizing the sum of squared residual of (9.28)

$$L(\beta) = \sum_{j=1}^{p} \|\mathbf{x}_j - \sum_{k \neq j} \beta_{jk} \mathbf{x}_k\|^2 \tag{9.29}$$

in a least squares fashion. If we denote the least squares estimator by $\widehat{\beta_{jk}}$, the residuals are given by

$$\mathbf{r}_j = \mathbf{x}_j - \sum_{k \neq j} \widehat{\beta_{jk}} \mathbf{x}_k. \tag{9.30}$$

The partial correlation is then obtained by computing the correlation between the residuals (Lerch et al., 2006; He et al., 2007; Peng et al., 2009):

$$\rho_{ij} = \text{corr}\,(\mathbf{r}_i, \mathbf{r}_j).$$

9.4.1 Sparse Partial Correlations

There is a serious problem with the least squares estimation framework discussed in the previous section. Since $n \ll p$, this is a significantly underdetermined system. This is also related to the covariance matrix Σ being singular so we cannot just invert the covariance matrix. For this, we need sparse network modeling.

The minimization of (9.29) is exactly given by solving the normal equation:

$$\mathbf{x}_j = \sum_{k \neq j} \beta_{jk} \mathbf{x}_k, \tag{9.31}$$

which can be turned into standard linear form $y = A\beta$ (Lee et al., 2011c). Note that (9.31) can be written as

$$\mathbf{x}_j = \underbrace{[\mathbf{x}_1, \cdots, \mathbf{x}_{j-1}, \mathbf{0}, \mathbf{x}_{j+1}, \cdots, \mathbf{x}_p]}_{\mathbf{X}_{-j}} \underbrace{\begin{pmatrix} \beta_{j1} \\ \beta_{j2} \\ \vdots \\ \beta_{jp} \end{pmatrix}}_{\beta_j},$$

where $\mathbf{0}_{n\times 1}$ is a column vector of all zero entries. Then we have

$$\underbrace{\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_p \end{pmatrix}}_{y_{np\times 1}} = \underbrace{\begin{pmatrix} \mathbf{X}_{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_{-2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{X}_{-p} \end{pmatrix}}_{A_{np\times p^2}} \underbrace{\begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}}_{\beta_{p^2\times 1}}, \tag{9.32}$$

where A is a block diagonal matrix and $\mathbf{0}_{n\times p}$ is a matrix of all zero entries. We regularize (9.32) by incorporating l_1 LASSO-penalty J (Tibshirani, 1996; Peng et al., 2009; Lee et al., 2011c):

$$J = \sum_{i,j} |\beta_{ij}|.$$

The sparse estimation of β_{ij} is then given by minimizing $L + \lambda J$. Since there is dependency between y and A, (9.32) is not exactly a standard compressed sensing problem (Peng et al., 2009; Lee et al., 2011c). It should be intuitively understood that sparsity makes the linear equation (9.31) less underdetermined. The larger the value of λ, the more sparse the underlying topological structure gets. Since

$$\rho_{ij} = \beta_{ij}\sqrt{\frac{\sigma^{ii}}{\sigma^{jj}}},$$

the sparsity of β_{ij} directly corresponds to the sparsity of ρ_{ij}, which is the strength of the link between nodes i and j (Peng et al., 2009; Lee et al., 2011c). Once the sparse partial correlation matrix ρ is obtained, we can simply link nodes j and j, if $\rho_{ij} > 0$ and assign the weight ρ_{ij} to the edge. This way, we obtain the weighted graph.

9.4.2 Limitations

However, the sparse partial correlation framework has a serious computational bottleneck. For n measurements over p nodes, it is required that we solve a linear system with an extremely large A matrix of size $np \times p^2$, so that the complexity of the problem increases by a factor of p^3! Consequently, for a large number of nodes, the problem immediately becomes almost intractable for a small computer. For example, for 1 million nodes, we have to compute 1 trillion possible pairwise relationships between nodes. One practical solution is to modify (9.28) so that the measurement at node i is represented more sparsely over some possible index set S_i:

$$x_i = \sum_{S_i} \beta_{ij} x_j + \epsilon_i.$$

making the problem substantially smaller.

An alternate approach is to simply follow the *homotopy path*, which adds network edges one by one with a very limited increase of computational complexity so there is no need to compute β repeatedly from scratch (Plumbley, 2005; Donoho and Tsaig, 2006; Osborne et al., 2000). The trajectory of the optimal solution β in LASSO follows a piecewise linear path as we change λ. By tracing the linear path, we can substantially reduce the computational burden of reestimating β when λ changes.

10

Brain Network Distances

Many existing brain network distances are based on matrix norms. The elementwise differences may fail to capture underlying topological differences. Further, matrix norms are sensitive to outliers. A few extreme edge weights may severely affect the distance. Thus it is necessary to develop network distances that recognize topology. In this chapter, we introduce Gromov–Hausdorff (GH) and Kolmogorov–Smirnov (KS) distances. GH distance is often used in persistent homology-based brain network models. In this chapter, various network distances are contrasted against each other in random network simulations with the ground truths.

There are many similarity measures and distances between networks in literature (Banks and Carley, 1994; Lee et al., 2012; Chung et al., 2015a; Chen et al., 2016). Many of these approaches simply ignore the topology of the networks and mainly use the sum of differences between either node or edge measurements. These network distances are sensitive to the topology of networks. They may lose sensitivity over topological structures such as the connected components, modules, and holes in networks. In standard graph theoretic approaches, the similarity and distance of networks are measured by determining the difference in graph theory features such as assortativity, betweenness centrality, small-worldness, and network homogeneity (Bullmore and Sporns, 2009; Uddin et al., 2008; Rubinov and Sporns, 2010). Comparison of graph theory features appears to reveal changes of structural or functional connectivity associated with different clinical populations (Rubinov and Sporns, 2010). Since weighted brain networks are difficult to interpret and visualize, they are often turned into binary networks by thresholding edge weights (He et al., 2008; Wijk et al., 2010). However, the choice of thresholding the edge weights may alter the network topology. To obtain the proper optimal threshold, the multiple comparison correction over every possible edge has been proposed (Salvador et al., 2005; Rubinov et al., 2009; Wijk et al., 2010). However, depending on what p-value to threshold, the resulting binary graph also changes. Others tried to control the sparsity of edges in the network in obtaining the binary network (Bassett, 2006; He et al., 2008; Wijk et al., 2010; Lee et al., 2011c). However, one encounters the problem of thresholding sparse parameters. Thus existing methods for binarizing weighted networks cannot escape the inherent problem of arbitrary thresholding.

Until now, there is no widely accepted criteria for thresholding networks. Instead of trying to come up with an optimal threshold for network construction that may not work for different clinical populations or cognitive conditions (Wijk et al., 2010), *why not use all networks for every possible threshold?* Motivated by this question, a new multiscale hierarchical network modeling framework based on persistent homology has been developed recently (Lee et al., 2011a,b, 2012; Chung et al., 2013, 2015a).

In persistent homology, there are various metrics that have been proposed to measure network distance. Among them, *Gromov–Hausdorff (GH) distance* is possibly the most popular distance that is originally used to measure distance between two metric spaces (Tuzhilin, 2016). It was later adapted to measure distances in persistent homology, dendrograms (Carlsson and Mémoli, 2010), and brain networks (Lee et al., 2012). The probability distributions of GH-distance is unknown. Thus, the statistical inference on GH-distance has been done through resampling techniques such as jackknife, bootstraps, or permutations (Lee et al., 2012, 2017; Chung et al., 2015a), which often cause computational bottlenecks for large-scale networks. To bypass the computational bottleneck associated with resampling large-scale networks, the *Kolmogorov–Smirnov (KS) distance* was introduced in (Chung, 2012; Chung et al., 2017a; Lee et al., 2017). The advantage of using KS-distance is its easiness to interpret compared to other less intuitive distances from persistent homology. Due to its simplicity, it is possible to determine its probability distribution exactly (Chung et al., 2017a).

Many distance or similarity measures are not metrics but having metric distances makes the interpretation of brain networks easier due to the triangle inequality. Further, existing network distance concepts are often borrowed from the metric space theory. Let us start with formulating networks as metric spaces.

10.1 Matrix Norms

Consider a weighted graph or network with the node set $V = \{1, \dots, p\}$ and the edge weights $w = (w_{ij})$, where w_{ij} is the weight between nodes i and j.

The edge weight is usually given by a similarity measure between the observed data on the nodes. Various similarity measures have been proposed. The correlation or mutual information between measurements for the biological or metabolic network and the frequency of contact between actors for the social network have been used as edge weights (Bassett, 2006; Bien and Tibshirani, 2011). We may assume that the edge weights satisfy the metric properties, nonnegativity, identity, symmetry, and the triangle inequality, such that

$$w_{i,j} \geq 0,\ w_{ii} = 0,\ w_{ij} = w_{ji},\ w_{ij} \leq w_{ik} + w_{kj}.$$

With theses conditions, $\mathcal{X} = (V, w)$ forms a metric space. Although the metric property is not necessary for building a network, it offers many nice mathematical properties and easier interpretation of network connectivity. Many real-world networks satisfy the metric properties.

Example 10.1 *Given measurement vector* $\mathbf{x}_i = (x_{1i}, \cdots, x_{ni})^\top \in \mathbb{R}^n$ *on the node* i*, the weight* $w = (w_{ij})$ *between nodes is often given by some bivariate function* f*:* $w_{ij} = f(\mathbf{x}_i, \mathbf{x}_j)$*. The correlation between* $\mathbf{x}_i$ *and* $\mathbf{x}_j$*, denoted as* $corr(\mathbf{x}_i, \mathbf{x}_j)$*, is a bivariate function. If the weights* $w = (w_{ij})$ *are given by*

$$w_{ij} = \sqrt{1 - corr(\mathbf{x}_i, \mathbf{x}_j)},$$

it can be shown that $\mathcal{X} = (V, w)$ *forms a metric space.*

Matrix norms of the difference between networks are often used as a measure of similarity between networks (Banks and Carley, 1994; Zhu et al., 2014). Given two networks $\mathcal{X}^1 = (V, w^1)$ and $\mathcal{X}^2 = (V, w^2)$, the L_l-norm of network difference is given by

$$D_l(\mathcal{X}^1, \mathcal{X}^2) = \| w^1 - w^2 \|_l = \Big(\sum_{i,j} \big| w_{ij}^1 - w_{ij}^2 \big|^l \Big)^{1/l}.$$

Note that L_l is the elementwise Euclidean distance in l-dimension. When $l = \infty$, L_∞-distance is written as

$$D_\infty(\mathcal{X}^1, \mathcal{X}^2) = \| w^1 - w^2 \|_\infty = \max_{\forall i,j} \big| w_{ij}^1 - w_{ij}^2 \big|.$$

The elementwise differences may not capture additional higher-order similarity. For instance, there might be relations between a pair of columns or rows (Zhu et al., 2014). Also L_1- and L_2-distances usually suffer the problem of outliers. Few outlying extreme edge weights may severely affect the distance. Further, these distances ignore the underlying topological structures. Thus, there is a need to define distances that are more topological.

Example 10.2 *Given two identical networks that only differ in one outlying link with infinite weight (Figure 10.1),* $L_l(\mathcal{X}^1, \mathcal{X}^2) = \infty$ *and* $L_\infty(\mathcal{X}^1, \mathcal{X}^2) =$

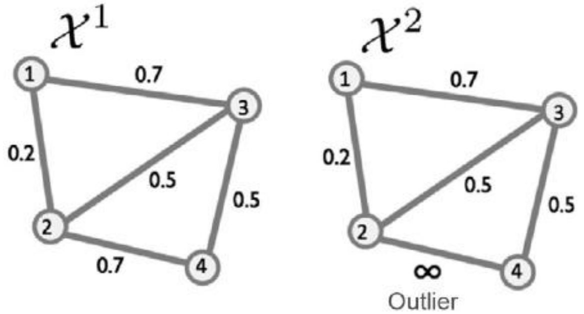


Figure 10.1 Toy networks with an outlying link weight. All the L_l-norm-based distance will give ∞ distance while GH- and KS-distance will give 0 distance.

∞. Thus, the usual matrix norm–based distance is sensitive to even a single outlying link. However, GH- and KS-distance is not sensitive to link weights but sensitive to the underlying topology. For $\mathcal{X}^1$ and $\mathcal{X}^2$, the corresponding single linkage matrices are identically

$$\begin{pmatrix} 0 & 0.2 & 0.5 & 0.5 \\ 0.2 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0 \end{pmatrix}.$$

GH-distance is the L_∞ norm on the single linkage matrix differences (Lee et al., 2012). Thus, GH-distance is 0 ignoring the outlier. The coordinates of the β_0 plot are given by $(0,1),(0.2,1),(0.5,1),(0.6,2),(0.7,2),(0.8,4)$ *for $\mathcal{X}^1$. For $\mathcal{X}^2$, the coordinates are all identical except* $(0.8,3)$*. Thus, KS-distance is 1 ignoring the influence of the outlier somewhat. Additional performance analysis on the poor performance of matrix norm–based distance is given in (Chung et al., 2017d).*

10.2 Bottleneck Distance

This is perhaps the most often used distance in persistent homology, but it is rarely used for brain networks. In persistent homology, the topology of underlying data can be represented by the birth and death of holes. In a graph, the zero- and one-dimensional holes are a connected component and a cycle, respectively (Carlsson et al., 2008). During a filtration, holes in a homology group appear and disappear. The Betti number at a particular threshold is then the number of holes at that threshold. If a hole appears at the threshold ξ and disappears at τ, it can be encoded into a point, (ξ, τ) $(0 \le \xi \le \tau < \infty)$ in $\mathbb{R}^2$.

If m number of holes appear during the filtration of a network $\mathcal{X} = (V, w)$, the homology group can be represented by a point set

$$\mathcal{P}(\mathcal{X}) = \{(\xi_1, \tau_1), \ldots, (\xi_m, \tau_m)\}.$$

This scatter plot is called the persistence diagram (PD) (Cohen-Steiner et al., 2007).

Given two networks $\mathcal{X}^1 = (V^1, w^1)$ with m holes and $\mathcal{X}^2 = (V^2, w^2)$ with n holes, we construct the corresponding graph filtrations. Subsequently, PDs

$$\mathcal{P}(\mathcal{X}^1) = \left\{(\xi_1^1, \tau_1^1), \cdots, (\xi_m^1, \tau_m^1)\right\}$$

and

$$\mathcal{P}(\mathcal{X}^2) = \left\{(\xi_1^2, \tau_1^2), \cdots, (\xi_n^2, \tau_n^2)\right\}$$

are obtained through the filtration (Lee et al., 2012). The bottleneck distance between the networks is defined as the bottleneck distance of the corresponding PDs (Cohen-Steiner et al., 2007):

$$D_B\big(\mathcal{P}(\mathcal{X}^1), \mathcal{P}(\mathcal{X}^2)\big) = \inf_{\gamma} \sup_{1 \le i \le m} \parallel t_i^1 - \gamma(t_i^1) \parallel_\infty,$$

where $t_i^1 = (\xi_i^1, \tau_i^1) \in \mathcal{P}(\mathcal{X}^1)$ and γ is a bijection from $\mathcal{P}(\mathcal{X}^1)$ to $\mathcal{P}(\mathcal{X}^2)$. The infimum is taken over all possible bijections. If $t_j^2 = (\xi_j^2, \tau_j^2) = \gamma(t_i^1)$ for some i and j, the L_∞-norm is given by

$$\parallel t_i^1 - \gamma(t_i^1) \parallel_\infty = \max\big(|\xi_i^1 - \xi_j^2|, |\tau_i^1 - \tau_j^2|\big).$$

Note that (10.1) assumes $m = n$ such that the bijection γ exists. Suppose two networks share the same node set, i.e., $V^1 = V^2$, with p nodes and the same number of q unique edge weights. If the maximal filtration is performed on two networks, the number of their zero- and one-dimensional holes that appear and disappear during the filtration is p and $1 - p + q$, respectively. Thus, their persistence diagrams of zero- or one-dimensional holes always have the same number of points. The bijection γ is determined by the bipartite graph matching algorithm (Cohen-Steiner et al., 2007; Edelsbrunner and Harer, 2008).

If $m \neq n$, there is no one-to-one correspondence between two PDs. Then, auxiliary points

$$\left(\frac{\xi_1^1 + \tau_1^1}{2}, \frac{\xi_1^1 + \tau_1^1}{2}\right), \ldots, \left(\frac{\xi_m^1 + \tau_m^1}{2}, \frac{\xi_m^1 + \tau_m^1}{2}\right)$$

and

$$\left(\frac{\xi_1^2 + \tau_1^2}{2}, \frac{\xi_1^2 + \tau_1^2}{2}\right), \ldots, \left(\frac{\xi_n^2 + \tau_n^2}{2}, \frac{\xi_n^2 + \tau_n^2}{2}\right)$$

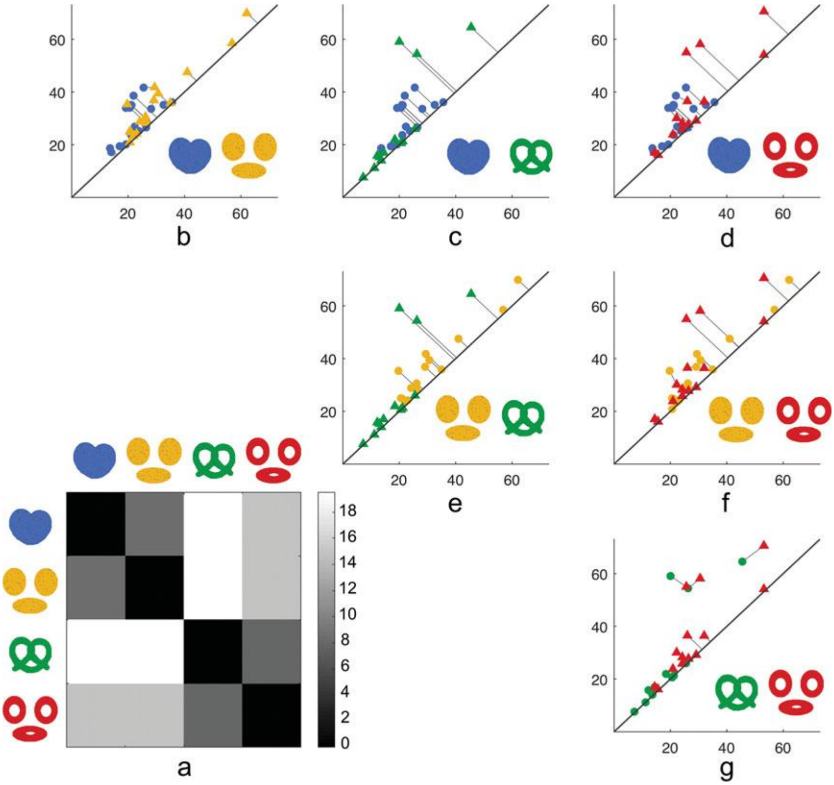
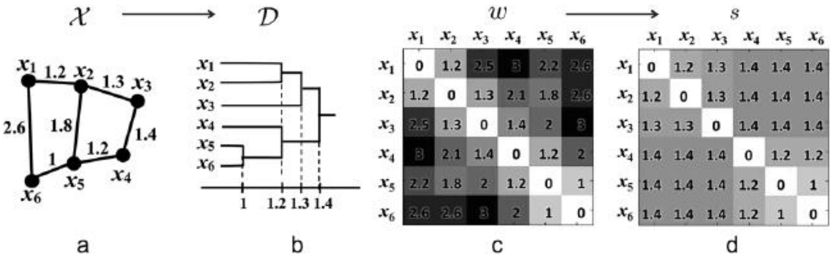


Figure 10.2 Toy example of bottleneck distance on four different types of scatter points: $\mathcal{X}^1$ (blue), $\mathcal{X}^2$ (yellow), $\mathcal{X}^3$ (green), and $\mathcal{X}^4$ (red). (a) Bottleneck distance displayed as a connectivity matrix (b–g) Persistence diagrams of holes of the pair of topological spaces. Blue, yellow, green, and red points correspond to the holes of blue, yellow, green, and red topological spaces, respectively. The figure was generated by Hyekyoung Lee of Seoul National University.

that are orthogonal projections to the diagonal line $\xi = \tau$ in $\mathcal{P}(\mathcal{X}^1)$ and $\mathcal{P}(\mathcal{X}^2)$ are added to $\mathcal{P}(\mathcal{X}^2)$ and $\mathcal{P}(\mathcal{X}^1)$ respectively to make the identical number of points in PDs (Figure 10.2).

10.3 Gromov–Hausdorff Distance

GH-distance for brain networks is first introduced in (Lee et al., 2012). GH-distance measures the difference between networks by embedding the network



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	1.2	2.5	3	2.2	2.6
x_2	1.2	0	1.3	2.1	1.8	2.6
x_3	2.5	1.3	0	1.4	2	3
x_4	3	2.1	1.4	0	1.2	2
x_5	2.2	1.8	2	1.2	0	1
x_6	2.6	2.6	3	2	1	0

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	1.2	1.3	1.4	1.4	1.4
x_2	1.2	0	1.3	1.4	1.4	1.4
x_3	1.3	1.3	0	1.4	1.4	1.4
x_4	1.4	1.4	1.4	0	1.2	1.2
x_5	1.4	1.4	1.4	1.2	0	1
x_6	1.4	1.4	1.4	1.2	1	0

Figure 10.3 (a) Toy network, (b) its dendrogram, (c) the distance matrix w based on Euclidean distance, and (d) the single linkage matrix (SLM) S. The figure was generated by Hyekyoung Lee of Seoul National University (Lee et al., 2011a).

into the ultrametric space that represents hierarchical clustering structure of network (Carlsson and Mémoli, 2010). The distance s_{ij} between the closest nodes in the two disjoint connected components $\mathbf{R}_1$ and $\mathbf{R}_2$ is called the single linkage distance (SLD), which is defined as

$$s_{ij} = \min_{l \in \mathbf{R}_1, k \in \mathbf{R}_2} w_{lk}.$$

Every edge connecting a node in $\mathbf{R}_1$ to a node in $\mathbf{R}_2$ has the same SLD. SLD is then used to construct the single linkage matrix (SLM) $S = (s_{ij})$ (Figure 10.3). SLM shows how connected components are merged locally and can be used in constructing a dendrogram. SLM is a *ultrametric*, which is a metric space satisfying the stronger triangle inequality $s_{ij} \leq \max(s_{ik}, s_{kj})$ (Carlsson and Mémoli, 2010). Thus the dendrogram can be represented as an ultrametric space $\mathcal{D} = (V, S)$, which is again a metric space.

Figure 10.4 shows the SLM corresponding to graphs:

$$\begin{pmatrix} 0 & 0.2 & 0.5 & 0.5 \\ 0.2 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 0.2 & 0.5 & 0.7 \\ 0.2 & 0 & 0.5 & 0.7 \\ 0.5 & 0.5 & 0 & 0.7 \\ 0.7 & 0.7 & 0.7 & 0 \end{pmatrix}.$$

GH-distance between networks is then defined through GH-distance between corresponding dendrograms. Given two dendrograms $\mathcal{D}^1 = (V, S^1)$ and $\mathcal{D}^2 = (V, S^2)$ with SLM $S^1 = (s^1_{ij})$ and $S^2 = (s^2_{ij})$,

$$D_{GH}(\mathcal{D}^1, \mathcal{D}^2) = \frac{1}{2} \max_{\forall i,j} |s^1_{ij} - s^2_{ij}|. \tag{10.1}$$

For the statistical inference on GH-distance, resampling techniques such as jackknife or permutation tests are often used (Lee et al., 2011a, 2012, 2017).

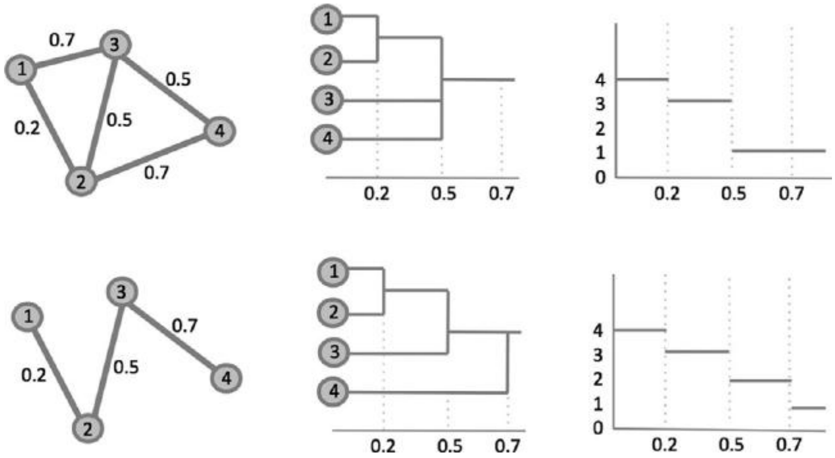


Figure 10.4 Toy networks with four nodes and corresponding dendrograms and β_0 plots.

In order to compare persistent homological structures including persistent diagrams and networks quantitively, it is necessary to define similarity or distance between objects. Numerous distances have been proposed including bottleneck, GH-, and KS-distances (Edelsbrunner and Harer, 2010; Lee et al., 2012; Chung et al., 2017a). We are using the distances themselves as test statistics, and they do provide intuitive topological interpretations.

GH-distance is the maximum of SLD differences (Lee et al., 2012). SLD between two nodes i and j is the shortest distance between two connected components that contain the nodes. The edges that give the maximum SLM is the GH-distance between the two networks.

Among all topological distances, only GH-distance uses the single linkage clustering. All other distances do not use the single linkage clustering (Chung et al., 2017d). For instance, bottleneck distance or KS-distance are not related to single linkage clustering (Lee et al., 2012).

10.4 Kolmogorov–Smirnov Distance

Recently KS-distance based on graph filtration has been proposed and successfully applied to brain networks as a way to quantify networks without thresholding (Chung et al., 2017a). The main advantage of the method is that the method avoids using the time-consuming permutation test for large-scale networks.

Definition 10.1 *Given weighted network $\mathcal{X} = (V, w)$ with edge weight $w = (w_{ij})$, the binary network $\mathcal{X}|_\epsilon = (V, w|_\epsilon)$ is a graph consisting of the node set V and the binary edge weights $w|_\epsilon$ given by*

$$w|_\epsilon = (w_{ij}|_\epsilon) = \begin{cases} 1 & \textit{if } w_{ij} > \epsilon; \\ 0 & \textit{otherwise.} \end{cases} \tag{10.2}$$

Note Lee et al. (2012) defines the binary graphs by thresholding above, which is consistent with the definition of the Rips filtration. However, in brain imaging, higher value w_{ij} indicates stronger connectivity. Thus, we are thresholding below (Chung et al., 2015a). With these notations, we define KS-distance as follows.

Given two networks $\mathcal{X}^1 = (V, w^1)$ and $\mathcal{X}^2 = (V, w^2)$, KS-distance between $\mathcal{X}^1$ and $\mathcal{X}^2$ is defined as (Chung et al., 2013, 2015a; Lee et al., 2017)

$$D_{KS}(\mathcal{X}^1, \mathcal{X}^2) = \sup_{\epsilon \geq 0} \left| f(\mathcal{X}^1_\epsilon) - f(\mathcal{X}^2_\epsilon) \right|$$

using monotone function f. The distance D_{KS} is motivated by the KS test for determining the equivalence of two cumulative distribution functions (Böhm and Hornik, 2010; Gibbons and Chakraborti, 2011; Chung et al., 2017a). The distance D_{KS} can be discretely approximated using the finite number of filtrations:

$$D_q = \sup_{1 \leq j \leq q} \left| f(\mathcal{X}^1|_{\epsilon_j}) - f(\mathcal{X}^2|_{\epsilon_j}) \right|.$$

If we choose enough number of q such that ϵ_j are all the sorted edge weights, then

$$D_{KS}(\mathcal{X}^1, \mathcal{X}^2) = D_q$$

(Chung et al., 2017a). This is possible since there are only up to $p(p-1)/2$ number of unique edges in a graph with p nodes and the monotone function increases discretely but *not continuously*. In practice, ϵ_j may be chosen uniformly or a divide-and-conquer strategy can be used to adaptively grid the filtration values.

The probability distribution of D_q under the null is asymptotically given by (Chung et al., 2017a)

$$\lim_{q \to \infty} \left(D_q / \sqrt{2q} \geq d \right) = 2 \sum_{i=1}^{\infty} (-1)^{i-1} e^{-2i^2 d^2}. \tag{10.3}$$

p-value under the null is then computed as

$$p\text{-value} = 2e^{-d_o^2} - 2e^{-8d_o^2} + 2e^{-18d_o^2} \cdots,$$

where the observed value d_o is the least integer greater than $D_q/\sqrt{2q}$ in the data. For any large value $d_0 \geq 2$, the second term is in the order of 10^{-14} and insignificant. Even for small observed d_0, the expansion converges quickly and five terms are sufficient. KS-distance method does not assume any statistical distribution of graph features other than that they must be monotonic. The technique is very general and applicable to other monotonic graph features such as node degrees.

Example 10.3 *Consider a network with edge weights $r_{ij} = 1 - corr(\mathbf{x}_i, \mathbf{x}_j)$. Such a network is not a metric space. To make it a metric space, we need to scale the edge weight to $w_{ij} = \sqrt{r_{ij}}$ (Example 10.1). However, KS-distance is invariant under such monotonic scaling since the distance is taken over every possible filtration value.*

Figure 10.5 displays the number of connected components (β_0), and the size of largest connected components (γ) in $\sqrt{1 - corr}$ plots of FA-values uniformly sampled at 1,856 nodes along the white matter template boundary for normal controls and maltreated children (Chung et al., 2015a, 2017b). This results in $1,856 \times 1,856$ correlation matrix for each group. Using the KS-distance, we determined the statistical significance of the correlation matrix differences between the groups. The statistical results in terms of p-values are all below 0.0001, indicating the very strong overall structural network differences in DTI.

Limitation of GH- and KS-distances. The limitation of the SLM is the inability to discriminate a cycle in a graph. Consider two topologically

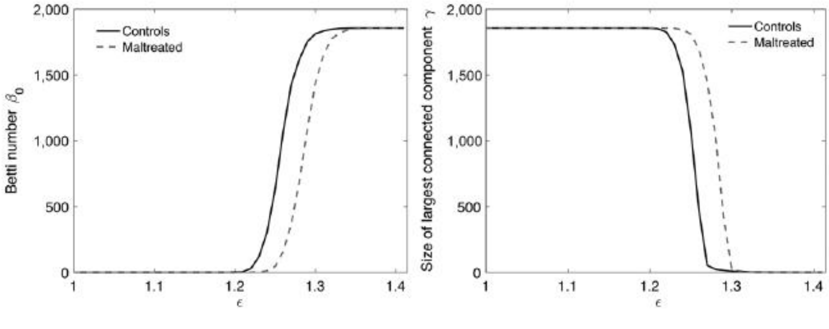


Figure 10.5 The plots of β_0 (left) and γ (right) over $\sqrt{1 - \text{corr.}}$ showing structural network differences between maltreated children (dotted line) and normal controls (solid line) on 1,856 nodes.

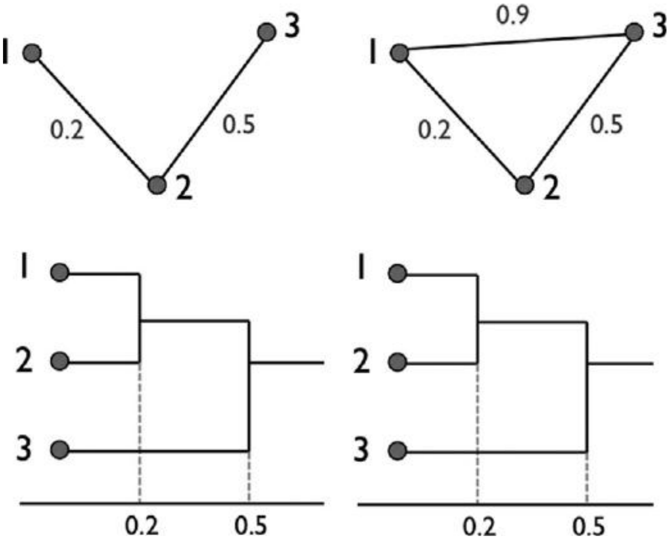


Figure 10.6 Two topologically distinct graphs may have identical dendrograms, which results in zero GH-distance.

different graphs with three nodes (Figure 10.6). However, the corresponding SLM are identically given by

$$\begin{pmatrix} 0 & 0.2 & 0.5 \\ 0.2 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 0.2 & 0.5 \\ 0.2 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{pmatrix}.$$

The lack of uniqueness of SLMs makes GH-distance incapable of discriminating networks with cycles (Chung, 2012). KS-distance also treats the two networks in Figure 10.6 as identical if Betti number β_0 is used as the monotonic feature function. Thus, KS-distance also fail to discriminate cycles. This example illustrates a need for new, more sophisticated network distance that can discriminate the higher-order topological features such as cycles and holes.

10.5 Performance Analysis

Different network distances introduced in the chapter are compared in two simulation studies involving the paired network settings (Chung et al., 2017a,d). The simulations are independently performed 100 times and the average results were reported.

10.5.1 Comparisons on Paired Networks

No network difference. There are three groups and the sample size is $n = 40$ in each group and the number of nodes are $p = 10$. In groups I and II, the kth data $x_k(v_i)$ at each node v_i was simulated as standard normal, i.e.,

$$x_k(v_i) \sim N(0, 1).$$

Additional data were simulated as

$$y_k(v_i) = x_k(v_i) + N(0, 0.01^2)$$

for all the nodes for groups I and II. The paired data $(x_1, y_1), \cdots (x_n, y_n)$ are then centered and scaled following (Chung et al., 2017a) and the cross-correlation between them is computed (Figure 10.7). Six different distances are used to determine if there are any network differences as expected. The results are given in Table 10.1. For the first four distances, permutation tests are used. Since there are five samples in each group, the total number of permutation is $\binom{10}{5} = 272$, making the permutation test exact and the comparison fair. All six distances performed reasonably well and did not detect network differences in average.

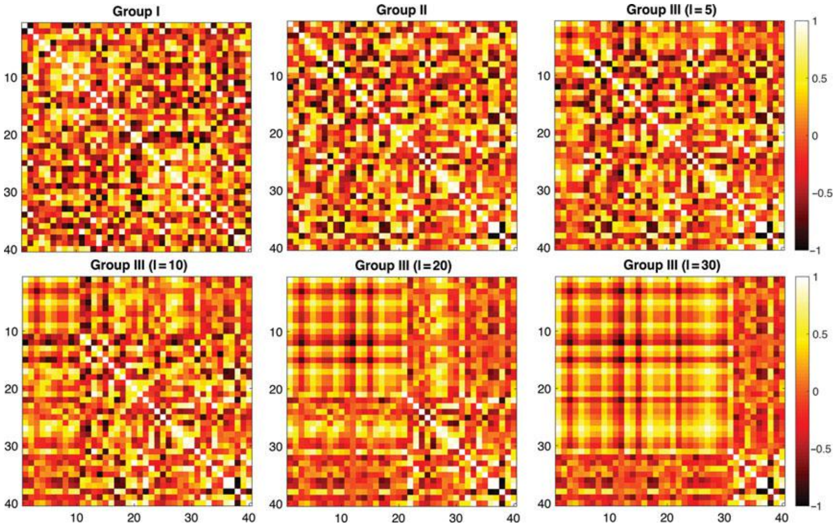


Figure 10.7 Randomly simulated correlation matrices. Group I and group II are generated independently and identically. Group III is generated independently but identically as group I but additional dependency is added for the first l nodes.

Table 10.1. *Simulation results given in terms of p-values. In the case of no network difference, higher p-values are better. In the case of network difference, smaller p-values are better. In the presence of signal, KS-like test procedures performed the best. The more structure (increased l) there is, KS-like tests performed substantially better.* * *indicates statististical significance below* 10^{-9}.

	L_1	L_2	L_∞
no diff.	0.48 ± 0.30	0.48 ± 0.30	0.52 ± 0.29
diff. ($l = 5$)	0.49 ± 0.29	0.51 ± 0.29	0.42 ± 0.30
diff. ($l = 10$)	0.47 ± 0.31	0.48 ± 0.31	0.37 ± 0.30
diff. ($l = 20$)	0.47 ± 0.31	0.51 ± 0.32	0.23 ± 0.26
diff. ($l = 30$)	0.38 ± 0.34	0.40 ± 0.35	0.11 ± 0.18

	GH	KS (β_0)	KS (γ)
no diff.	0.53 ± 0.30	0.64 ± 0.33	0.28 ± 0.38
diff. ($l = 5$)	0.19 ± 0.24	0.38 ± 0.29	0.17 ± 0.28
diff. ($l = 10$)	0.12 ± 0.16	0.05 ± 0.12	0.04 ± 0.13
diff. ($l = 20$)	0.07 ± 0.08	0.00 ± 0.00	0.00 ± 0.00
diff. ($l = 30$)	0.05 ± 0.06	$(0.00 \pm 0.00)^*$	$(0.00 \pm 0.00)^*$

Network difference. Group III was generated identically and independently like group I, but additional dependency was added by letting $y_k(v_j) = x_k(v_1)/2$ for l nodes indexed by $j = 1, 2, \cdots, l$. We changed the number of dependent nodes l to be 5, 10, 20 and 30. This dependency gives high-connectivity differences between groups I and III (Figure 10.7). When the number of dependencies is low ($l = 5$), all the methods performed badly. However, as the number of dependency increases, the performance of L_∞, GH-distance, and KS-like distances increased. In particular, KS-like distance substantially outperformed when there are many dependent nodes ($l = 20, 30$).

In terms of computation, distance methods based on the permutation test took about 950 seconds (16 minutes) while the KS-like test procedure only took about 20 seconds on a computer.

10.6 Comparisons on Modules

Five different network distances (L_1, L_2, L_∞, GH, and KS) were compared in simulation studies with modular structures. There were four groups, the

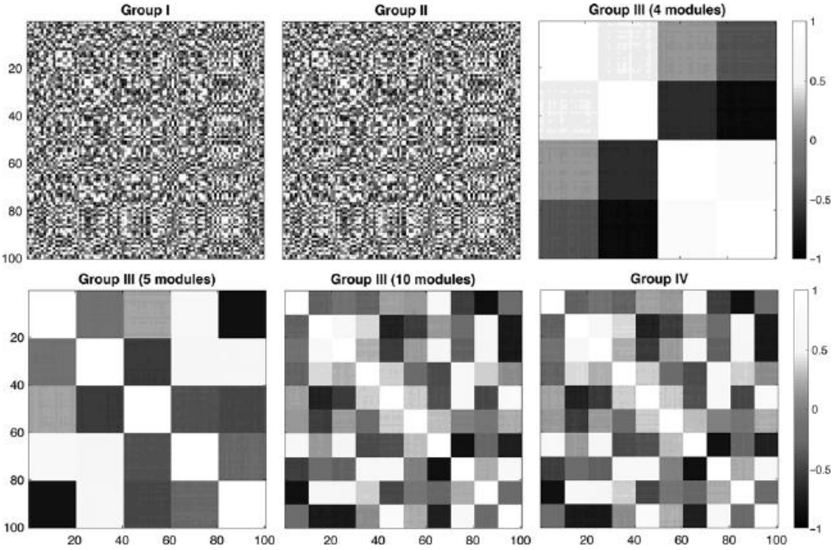


Figure 10.8 Randomly simulated correlation matrices. Group I and group II were generated independently and identically. Group III was generated from group I but additional dependency was added to introduce modular structures. Group IV was generated from group III (10 modules) by adding small noise.

sample size was $n = 5$ in each group and the number of nodes was $p = 100$ (Figure 10.8). We follow notations in Example 10.1. In group I, the measurement vector $\mathbf{x}_i$ at node i was simulated as multivariate normal, i.e., $\mathbf{x}_i \sim N(0, I_n)$ with n by n identity matrix I_n as the covariance matrix. The edge weights for group I was

$$w^1_{ij} = \sqrt{1 - \mathrm{corr}(\mathbf{x}_i, \mathbf{x}_j)}.$$

In group II, the measurement vector $\mathbf{y}_i$ at node i was simulated as

$$\mathbf{y}_i = \mathbf{x}_i + N(0, \sigma^2 I_n)$$

with noise level $\sigma = 0.01$. The edge weight for group II was

$$w^2_{ij} = \sqrt{1 - \mathrm{corr}(\mathbf{y}_i, \mathbf{y}_j)}.$$

Group III was generated by adding additional dependency to group I:

$$\mathbf{y}_i = 0.5\mathbf{x}_{ci+1} + N(0, \sigma I_n).$$

This introduces the modules in the network. We assumed there were total $k = 4, 5, 10$ modules and each module consists of $c = p/k$ number of points. Group IV was generated by adding noise to group III: $\mathbf{z}_i = \mathbf{y}_i + N(0, \sigma^2 I_n)$.

Table 10.2. *Simulation results given in terms of p-values. In the case of no network differences (0 vs. 0 and 4 vs. 4), higher p-values are better. In the case of network differences (4 vs. 5 and 5 vs. 10), smaller p-values are better. * and ** indicates multiplying* 10^{-3} *and* 10^{-4}.

	L_1	L_2	L_∞
0 vs. 0	0.93 ± 0.04	0.93 ± 0.04	0.93 ± 0.04
4 vs. 4	0.89 ± 0.02	0.89 ± 0.02	0.90 ± 0.03
4 vs. 5	0.14 ± 0.16	0.06 ± 0.10	0.03 ± 0.06
5 vs. 10	0.47 ± 0.25	0.19 ± 0.18	0.10 ± 0.10

	GH	KS (β_0)	KS (γ)
0 vs. 0	0.87 ± 0.14	1.00 ± 0.00	1.00 ± 0.00
4 vs. 4	0.86 ± 0.17	0.87 ± 0.29	0.88 ± 0.28
4 vs. 5	0.29 ± 0.30	$(0.07 \pm 0.67)^{**}$	$(0.07 \pm 0.67)^{**}$
5 vs. 10	0.33 ± 0.30	0.01 ± 0.08	$(0.06 \pm 0.53)^{*}$

No network difference. It was expected there was no network difference between groups I and II. We applied the five different distances. For the first four distances, a permutation test was used. Since there were five samples in each group, the total number of permutations was $\binom{10}{5} = 272$, making the permutation test exact and the comparisons fair. All the distances performed well and did not detect network differences (Table 10.2). It was also expected there is no network difference between groups III and IV. We compared four-module network to four-module network. All the distances performed equally well and did not detect differences (Table 10.2).

Network difference. Networks with four, five, and 10 modules were generated using group III models. Since the number of modules was different, they were considered as different networks. We compared four- and five-module networks, and five- and 10-module networks (Table 10.2). L_1, L_2, L_∞ distances did not perform well for five. vs. 10 module comparisons. Surprisingly, GH-distance performed worse than L_∞ in all cases. On the other hand, KS-distance performed extremely well.

The results of the aforementioned simulations did not change much even if we increased the noise level to $\sigma = 0.1$. In terms of computation, distance methods based on the permutation test took about 950 seconds (16 minutes), while the KS-like test procedure only took about 20 seconds on a computer.

The MATLAB code for performing these simulations is provided.[1] The results given in Table 10.2 may slightly change if different random networks are generated.

10.7 Hypernetworks

Large-scale manifold-valued data such as a collection of correlation matrices and networks have recently attracted substantial attention in the scientific community, in areas such as neuroscience, environmental studies, and sociology, where the networks are of prime interest (Tenenbaum et al., 2000; Fletcher et al., 2004). In modeling such complex data, it is crucial to be able to visualize the relationship information such as dependence and similarity between manifold-value data. Since such data are tensorial with a very complex geometric structure, it is not straightforward to compute and visualize relationship information. In the study of brain connectivity, networks represented as manifold-valued data have far more complex geometric and topological structures than the usual Euclidean geometry admits. It is not even clear how to characterize the distributions of manifold-valued data. The hypernetwork approach can be used to empirically estimate the center and spread of the distributions, as well as discover associations and relationships between manifold-valued data. These empirical approaches will be particularly useful for the study of brain connectivity, where the characterization of variability between subjects and over time remains a challenging problem.

The *hypernetwork* can be used to integrate multiple measures of manifold-valued data from different time points, groups, trials, experiments, sites, and modalities. The hypernetwork can integrate a collection of manifold-valued data in a holistic fashion as a single gigantic network, where a node is a manifold-value data point and the edge is the distance or similarity measure between manifold-value data points; thus a hypernetwork is a *network of networks* (Chung et al., 2017a). Although hypernetworks are frequently used in machine learning (Zhang, 2008), the concept has not often been applied in neuroimaging or other areas of science. We will treat the brain network of a single subject as a single point in higher-dimensional abstract space. Then we connect all the points to form a hypernetwork. In a hypernetwork, a node is a network, and an edge is the distance between networks. The hypernetwork can be used to study the distribution of brain connectivity patterns across individuals.

[1] http://www.stat.wisc.edu/~mchung/twins/

10.7.1 Hypernetwork of Manifold-Valued Data

We can use the log-Euclidean or other matrix-norm based distances for edge weights in hypernetworks (Chung et al., 2017d). Often correlation between two correlation matrices are used as distance between two correlation matrices. In Zhang et al. (2016) and Zhao et al. (2018), correlation of correlations are used to capture high-order functional interactions across brain regions. We can also use the topological distance that is topologically invariant for measuring distance between networks (Chung et al., 2017d). This requires determining how close two graphs are topologically. This is still a very challenging open methodological problem in the field. Alternately, we can compute the distance between graphs using the topological distances such as GH and bottleneck distances, which are deformation-invariant similarity measures often used in persistent homology, but rarely used in brain networks (Mémoli, 2008; Lee et al., 2011a, 2012). KS-distance, which was recently proposed as an alternative to GH or bottleneck distances, can be used. The main advantage of topological distances over geometric distances is that we can measure distance between networks that do not have the same number of nodes or anatomical correspondence between nodes. Thus, it is possible to measure distance between EEG and fMRI networks with different numbers of nodes.

Given connectivity $Y_1, \cdots, Y_n$ in metric space $\mathcal{M}$ with metric d, which measures the distance between data points. For instance, in the space of $p \times p$ positive definite symmetric (PDS) matrices $\mathcal{S}_p$, the geodesic is given by log-Euclidean framework (Moakher, 2005; Arsigny et al., 2006):

$$d(Y_i, Y_j) = \|\log(Y_i^{-1/2} Y_j Y_i^{-1/2})\|,$$

where $\| \cdot \|$ is the Euclidean norm. Then the hypernetwork of $Y_1, \cdots, Y_n$ consists of n nodes where the ith node is Y_i and edge weights between them is given by $d(Y_i, Y_j)$. Figure 10.9 illustrates a schematic of a hypernetwork obtained from a collection of brain networks consisting of PDS matrices. This results in an $n \times n$ connectivity matrix $D = (d(Y_i, Y_j))$, which is not intuitive to understand or easy to visualize. Thus, it is necessary to embed the hypernetwork into a 2D plane and 3D volume in such a way that the relative distances $d(Y_i, Y_j)$ are preserved (Tenenbaum et al., 2000). This will give an isomap embedding for visual display.

To obtain an overall smooth pattern of hypernetwork, we will perform iterated local weighted averaging (Chung et al., 2005b,a) such that all sample points Y_i will be updated by

$$Y_i \leftarrow \sum_{j \in N_i} w_{ij} Y_j,$$

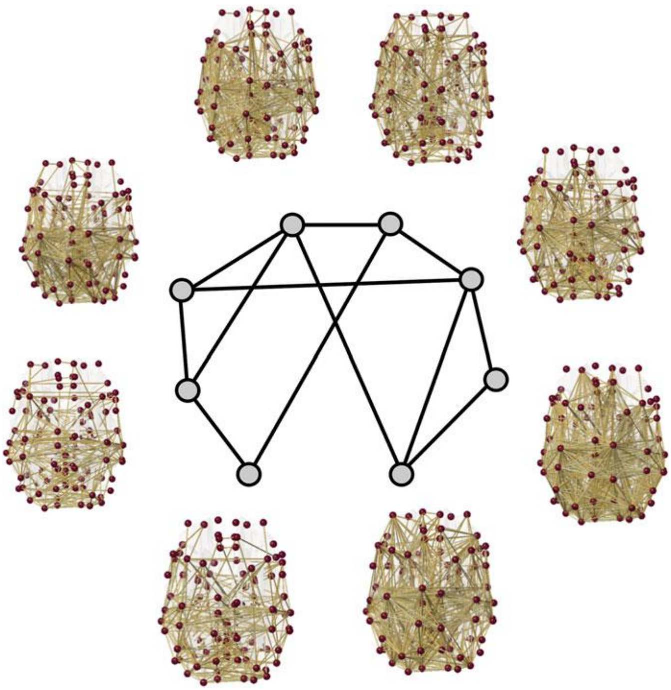


Figure 10.9 Schematic of hypernetwork construction between eight individual brain networks Y_i. The distance between the networks $d(Y_i, Y_j)$ is computed using the GH-distance. Edges with GH-distance larger than the prespecified threshold is removed for better biological interpretation.

where N_i is the neighboring points of Y_i determined by any point Y_j that is smaller than $d(Y_i, Y_j) \leq \epsilon$, and for weights $w_{ij} \propto \exp^{-d^{-1}(Y_i, Y_j)}$, which sums to 1. Note the updated Y_i are still PDS since $\mathcal{S}_p$ is convex. If we repeat this process a sufficient number of times, a steady-state pattern, the center will emerge. We can show that the steady-state pattern is related to the Fréchet mean, which is defined as

$$\mu_F = \arg\min_{Y \in \mathcal{S}_p} \sum_{j=1}^{n} d^2(Y, Y_j).$$

The *spread* of the pattern of the hyper network will be determined as the number of iterations that will reach the steady state, which is related to the Fréchet variance, defined as

$$\sigma_F^2 = \frac{1}{n} \sum_{j=1}^{n} d^2(\mu_F, Y_j).$$

It is possible to identify conditions on ϵ and the number of iterations that will provide meaningful center and spread for the multitude of the data set. It is possible to use other distance measures on $\mathcal{S}_p$ beyond the log-Euclidean framework, such as Gaussian kernel weighting and generalized correlation measures.

10.7.2 Clustering in Hypernetwork

To discover association and relationship between manifold-valued data, we can perform hierarchical clustering. k-means is a robust and widely used method for unsupervised clustering in Euclidean (Moakher and Zéraï, 2011; Prabhu and Anbazhagan, 2011). For manifold-valued data, it may not be the most efficient choice, since the Euclidean distance ignores the underlying geometric structure of manifolds. Further, k-means does not yield hierarchically nested structure over k, which makes interpretation of the result difficult in the exploratory stage. However, it is possible to build hierarchical k-means clustering (Arai and Barakbah, 2007). The k-means function can be adapted to incorporate geodesics on $\mathcal{S}_p$:

$$J = \sum_{j=1}^{k} \sum_{i \in C_j} d^2(Y_i, \mu_j),$$

where μ_j is the Fréchet mean of the jth cluster C_j. Then using the Fréchet means $\mu_1, \cdots, \mu_k$ as the next seed points, we apply k-means clustering again with the smaller number of clusters. This process continues until we obtain the desired number of hierarchy. It is known that PCA in both the temporal (Ting et al., 2018) and spectral domains (Wang et al., 2016b) may improve the performance and stability of classical k-means clustering (Ding and He, 2004). It may be possible to extend PCA and independent components analysis (ICA) to incorporate the geodesic structure of the manifold data (Fletcher et al., 2004) to improve the performance of the hierarchical clustering. Alternately, one may construct the Laplacian of the hypernetwork and perform spectral clustering.

10.7.3 Graph Theory on Hypernetwork

Given p graphs $G_1, G_g, \cdots, G_p$ suppose we computed the pairwise distance $D = (d_{ij}) = d(G_i, G_j)$, which serves the edge weights in a hypernetwork. Using graph theory features, it is possible to quantify various properties of hypernetworks as a function of distances (Hagmann et al., 2008; Lee et al., 2018b).

Then the average pairwise distance is given by

$$\bar{d} = \frac{2}{p(p-1)} \sum_{i>j} d_{ij},$$

which is equivalent to the average node degree

$$\frac{1}{p(p-1)} \sum_{i=1}^{p} \sum_{i \in N_i} d_{ij},$$

where N_i is the set of nodes containing the neighbors of i but excluding i. The *average efficiency* of the hypernetwork is given by

$$E = \frac{1}{p(p-1)} \sum_{i>j} \frac{1}{d_{ij}}.$$

The efficiency of a network measures how efficient information is transferred between nodes. Smaller the distance between nodes, faster the information transfers.

11
Combinatorial Inferences for Networks

Permutation testing is known as the only exact test procedure in statistics (Nichols and Holmes, 2002; Chung et al., 2017a). However, it is not exact in practice and only an approximate method due to the computational bottleneck of enumerating every possible permutation. When dealing with big imaging data with large samples, it may take many years to do permutation in a desktop computer for converging results. In this chapter, we go explain how to use permutation test for brain network inferences. Then we will show how to do an exact permutation test using a new combinatorial inference procedure.

11.1 Permutation Test

The permutation test is perhaps the most widely used nonparametric test procedure in brain network inference (Zalesky et al., 2010; Chung et al., 2017a). It is known as the only exact test in statistics since the distribution of the test statistic under the null hypothesis can be exactly computed by calculating all possible values of the test statistic under every possible permutation (Fisher, 1966; Gibbons and Chakraborti, 2011; Chung et al., 2017a).

It is often used when parametric assumptions are in doubt or very difficult to obtain. Decades ago, the computational bottleneck of generating empirical distribution was so overwhelming, it was mainly applied to scalar data sets with small sample sizes. In recent years, due to the advancement of relatively inexpensive computers and the advancement of cluster computing, the permutation test became practical for a wide range of problems. However, it is still computationally expensive for even modest sample sizes used in most brain imaging studies. For really large-scale data sets, it is still computationally intensive.

To speed up the permutation tests when the total number of permutations is too large, resampling technique has been proposed under many different names, such as approximate permutation test or random permutation test. In the resampling technique, only a small fraction of possible permutations are generated and the statistical significance is approximately computed. This approximate permutation test is the most widely used version of the permutation test in brain imaging. Even for modest sample sizes, the total number of permutations is astronomically large and only a small subset of permutations is used in approximating p-values. In most of brain imaging studies, 5,000 to 1,000,000 permutations are often used, which puts the total number of permutations usually at less than 1% of all possible permutations. In Zalesky et al. (2010), 5,000 permutations out of a possible $\binom{27}{12} = 17,383,860$ permutations (2.9%) were used. In Thompson et al. (2001), 1 million permutations out of $\binom{40}{20}$ possible permutations (0.07 %) were used. In Lee et al. (2017), 5,000 permutations out of a possible $\binom{33}{10} = 92,561,040$ permutations (0.005%) were used.

Thus, permutation tests are all approximate in practice. Here, we propose a novel exact combinatorial test procedure that enumerates all possible permutations combinatorially and avoids the numerical resampling that is slow and approximate. Unlike existing permutation testing that takes a few hours to a day, our exact procedure takes a few seconds. Recently, combinatorial approaches for network inference are begin to emerge as the powerful alternative to existing standard network inference procedure (Neykov et al., 2016; Chung et al., 2017a). Neykov et al. proposed to use a combinatorial technique for graphical models. However, their approach still relies on bootstrapping. Thus it is still an approximate resampling method (Neykov et al., 2016). Chung et al. proposed a combinatorial approach for large-scale brain networks, but the method is limited to integer-valued graph theoretic features such as the number of connected components (Chung et al., 2017a).

11.1.1 Permutations

Definition 11.1 *Given n unique objects* $a_1, a_2, \cdots, a_n$*, a* permutation *is a rearrangement of the n objects. Given n unique objects* $a_1, a_2, \cdots, a_n$*, a k-combination is a way of selecting k distinct objects out of n total objects.*

For n unique objects $a_1, a_2, \cdots, a_n$, one possible permutation is

$$a_2, a_1, a_3, \cdots, a_n.$$

For n unique objects, the total number of possible permutations is

$$n! = n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1.$$

For given objects a_1, a_2, a_3, a_4, possible two-combinations are

$$(a_1, a_2), (a_1, a_3), (a_1, a_4), (a_2, a_3), (a_2, a_4), (a_3, a_4).$$

There are six different two-combinations.

Theorem 11.1 *The total number of possible k-combinations equals the* binomial coefficient

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}.$$

Proof. This can be seen by determining the relationship between k-combinations and permutation of n objects. If we let $\binom{n}{k}$ be the total number of k-combinations, there are $k!$ ways to permute within each combination. $\binom{n}{k}k!$ is the number of ways of picking out k objects out of n objects and order them. This is equivalent to

$$n(n-1)(n-2)\cdots(n-k+1) = \frac{n!}{(n-k)!}.$$

Thus,

$$\binom{n}{k}k! = \frac{n!}{(n-k)!}. \quad \square$$

Similarly we can count the number of possible combinations in the two samples. Suppose there are m unique objects $a_1, a_2, \cdots, a_m$ in group I and n unique objects $b_1, b_2, \cdots, b_n$ in group II. Combine them and split them randomly into n and m objects. We are not interested in ordering. The possible ways of the random splits is given by

$$\binom{m+n}{m} = \frac{(m+n)!}{m!\,n!}.$$

Based on Stirling's formula,

$$q! \sim \sqrt{2\pi q}\left(\frac{q}{e}\right)^q,$$

the total number of permutations in permuting two groups of size n each is

$$\binom{2n}{n} \sim \frac{4^n}{\sqrt{2\pi n}},$$

which becomes extremely large for even modest n. This is why permutation tests are exponentially slow.

Here are few examples of the number of combinations computed using MATLAB.

```
format long
nchoosek(20,10)
ans =
      184756

nchoosek(40,20)
ans =
   1.378465288200000e+11

nchoosek(80,40)

Warning: Result may not be exact.
Coefficient is greater than 9.007199e+15
and is only accurate to 15 digits

ans =
   1.075072087333362e+23
```

If there is $n = 40$ samples in each group, this is beyond MATLAB's capability and will not compute the number of permutations exactly. One way of handling a large number on computer is to use logarithms. Note

$$n! = e^{\sum_{j=1}^{n} \log j}.$$

```
format long
sum(log(1:80)) - 2*sum(log(1:40)))
ans =
 53.031844856178935
```

$\binom{80}{40}$ is computed as $e^{53.031844856178935}$:

11.1.2 Permutation Test on Two Samples

The permutation test (Fisher, 1966) is perhaps the most widely used nonparametric test procedure in medical imaging. It is known as the only *exact test* since the distribution of the test statistic under the null hypothesis can be exactly computed by calculating all possible values of the test statistic under every possible combinations. Unfortunately, even for modest sample sizes, the total number of permutations is astronomically large and only a small subset of permutations is used in approximating p-values. Thus, permutation tests are all approximate in practice.

The permutation test for two samples is done as follows (Efron, 1982; Lee et al., 2012; Chung et al., 2013). A two-sample test setting is probably the most often used in applications. Suppose there are m unique objects $a_1, a_2, \cdots, a_m$ in group I and n unique objects $b_1, b_2, \cdots, b_n$ in group II. Let $X^1 = (a_1, a_2, \cdots, a_m)$ and $X^2 = (b_1, b_2, \cdots, b_n)$. We are interested in testing if

$$H_0 : \text{Group I} = \text{Group II vs. } H_1 : \text{Group I} \neq \text{Group II}.$$

Let $D(X^1, X^2)$ be the test statistic that measures the distance between X^1 and X^2. We can simply use the sample mean difference

$$D(X^1, X^2) = \bar{a} - \bar{b}$$

or its normalized version, i.e., t-statistic, as the test statistic. Any reasonable distance function can be used for this purpose.

We test the statistical significance of distance $D(X^1, X^2)$ under the null hypothesis H_0. This requires knowing the probability distribution of D under null. Under the null hypothesis, two groups are interchangeable and $D(X^1, X^2)$ is expected to be close to 0. Thus, the sample space is generated as by permutations. The permutations are done as follows. Concatenate data $X = (X^1, X^2)$. Now permute the indices of X in the symmetric group of degree $m + n$, i.e., S_{m+n} (Kondor et al., 2007). The permuted data matrix is denoted as X_σ, where $\sigma \in S_{m+n}$. Then we split X_σ into two parts

$$X_\sigma = [X_\sigma^1, X_\sigma^2],$$

where X_σ^1 and X_σ^2 are of sizes m and n respectively. Then for each permutation, we have distance $D(X_\sigma^1, X_\sigma^2)$. Theoretically, for all S_{m+n}, we can compute the distances. This gives the empirical estimation of the distribution of $D(X_\sigma^1, X_\sigma^2)$. The number of permutations exponentially increases and it is impractical to generate every possible permutation. So up to tens of thousands permutations are generated in practice. This is an approximate method and care should be taken to guarantee the convergence.

Given three groups with n, m, and l objects, we can also design the permutation test for three groups and test for the equality of the group means. However, due to the astronomically large number of permutations for three groups, the procedure simply causes the computational bottleneck. The permutation test is rarely applied in three groups.

11.1.3 Permutation Test on Correlations

Let $\rho_1(p)$ and $\rho_2(p)$ be the *population-level* correlations from two groups at each voxel p. We are interested in testing

$$H_0 : \rho_1(p) = \rho_2(p) \text{ for all } p$$

vs.

$$H_1 : \rho_1(p) \neq \rho_2(p) \text{ for some } p.$$

Let r_j be the Pearson correlation for the jth group. We can use

$$D = \sup_p \big[r_1(p) - r_2(p)\big]$$

as a test statistic. The test statistic will correct for multiple comparisons in a one-sided test. We estimate the probability distribution of D under H_0 via permutations. Each permutation produces Pearson correlations denoted by r_{1_σ} and r_{2_σ} and there are $\binom{m+n}{m}$ possible permutations. For each permutation, we compute distance D_σ. Then the distribution of the D is given by

$$P\Big(D \geq h\Big) = \frac{\#\text{ of } D_\sigma \geq h}{\binom{m+n}{m}}, \tag{11.1}$$

where # counts the number of instances when D_σ is bigger than h.

In order to have converging probability (11.1), we need to have tens of thousands of permutations (Figure 11.1). However, in specific situations, the permutation can be done by combinatorial enumerations exactly.

11.1.4 Permutation Test on Network Distances

Statistical inference on network distances can be done using the permutation test or bootstrap (Efron, 1982; Lee et al., 2012; Chung et al., 2013). Here we explain the permutation test procedure that was used for network distances. The usual setting in brain imaging applications is a two-sample comparison. Suppose there are m measurement in group 1 on node set V of size p. Denote the data matrix as $\mathbf{X}^1_{m\times p}$. The edge weights of group 1 are given by $f(\mathbf{X}^1)$ for some function f and the metric space is given by $\mathcal{X}^1 = (V, f(\mathbf{X}^1))$. Suppose there are n measurement in group 2 on the identical node set V. Denote data matrix as $\mathbf{X}^2_{n\times p}$ and the corresponding metric space as $\mathcal{X}^1 = (V, f(\mathbf{X}^1))$. We test the statistical significance of network distance $D(\mathcal{X}^1, \mathcal{X}^2)$ under the null hypothesis H_0:

$$H_0 : D(\mathcal{X}^1, \mathcal{X}^2) = 0 \text{ vs. } H_1 : D(\mathcal{X}^1, \mathcal{X}^2) \neq 0.$$

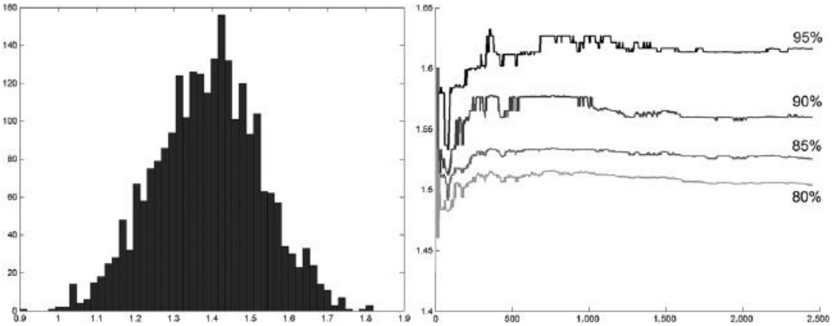


Figure 11.1 Left: histogram of distance D based on 2,400 permutations using $m = 14$ autistic and $n = 12$ normal control subjects for cortical thickness (Chung et al., 2005b). Right: plots of 95%, 90%, 85%, and 80% upper percentiles over the number of permutations showing that the p-values do not converge with less than 2,000 permutations. We most likely need more than 10,000 permutations for reasonable convergence.

The permutation test is done as follows. Concatenate the data matrices

$$\mathbf{X} = (x_{ij}) = \begin{pmatrix} \mathbf{X}^1 \\ \mathbf{X}^2 \end{pmatrix}_{(m+n)\times p}.$$

Now permute the indices of row vectors of $\mathbf{X}$ in the symmetric group of degree $m+n$, i.e., S_{m+n} (Kondor et al., 2007). The permuted data matrix is denoted as $\mathbf{X}_{\sigma(i)} = (x_{\sigma(i),j})$, where $\sigma \in S_{m+n}$. Then we split $\mathbf{X}_{\sigma(i)}$ into submatrices such that

$$\mathbf{X}_{\sigma(i)} = \begin{pmatrix} \mathbf{X}^1_{\sigma(i)} \\ \mathbf{X}^2_{\sigma(i)} \end{pmatrix},$$

where $\mathbf{X}^1_{\sigma(i)}$ and $\mathbf{X}^2_{\sigma(i)}$ are of sizes $m \times p$ and $n \times p$ respectively. Let $\mathcal{X}^1_{\sigma(i)} = (V, f(\mathbf{X}^1_{\sigma(i)}))$ and $\mathcal{X}^2_{\sigma(i)} = (V, f(\mathbf{X}^2_{\sigma(i)}))$ be weighted networks where the rows of the data matrices are permuted across the groups. Then we have distance $D(\mathcal{X}^1_{\sigma(i)}, \mathcal{X}^2_{\sigma(i)})$ for each permutation. The number of permutations exponentially increases and it is impractical to generate every possible permutation. So up to tens of thousands permutations are generated to guarantee convergence in practice. This is an approximate method and a care should be taken to guarantee the convergence, but in most studies about 1% of total permutations are used (Thompson et al., 2001; Zalesky et al., 2010).

11.2 Exact Combinatorial Inference

Definition 11.2 *For any G_1 and G_2 satisfying $G_1 \subset G_2$, f is a monotone function if it satisfies $f(G_1) < f(G_2)$.*

There are many monotone functions, including the number of connected components, total node degree, and the size of the largest connected components. Once we have a set of features that are monotone, statistical inference can be done easily. The method works best if we have integer-valued monotone graph theory features without any gap in integer values. This is defined more succinctly as follows.

Definition 11.3 *Given a collection of nested sets $\varnothing = G_0 \subset G_1 \subset G_2 \subset \cdots \subset G_q$, f is* monotonically dense *if*

$$0 = f(G_0) < f(G_1) < f(G_2) < \cdots < f(G_q) = r$$

and $r \leq q$.

Due to the pigeonhole principle, the integers between 0 and r are completely covered by $q + 1$ features.

Example 11.1 *Given graph G with q nodes $v_1, v_2, \cdots, v_q$, define G_j to be a subgraph with nodes $v_1, v_2, \cdots, v_j$. Let $f(G_j)$ to be the number of nodes in G_j. Then $f(G_j) = j$ and satisfies the condition in Definition 11.3 trivially.*

Even if monotone function f does not satisfy Definition 11.3, there exists a nondecreasing function ϕ such that $\phi \circ f$ is monotonically dense. Such function ϕ is easily constructed as follows. Let $x_j = f(G_j)$. Define a step function ϕ such that

$$\phi(t) = \begin{cases} 0 & \text{if } t < x_1 \\ j & \text{if } x_j \leq t < x_{j+1} \\ q & \text{if } x_q \leq t \end{cases} .$$

Then it is straightforward to see that $\phi \circ f$ is monotonically dense. An example of such a step function is illustrated in Figure 11.2. From now on, without loss of generality, we will simply assume a monotone feature to be monotonically dense.

We are interested in testing the null hypothesis H_0 of the equivalence of two monotonically dense features, f and g:

$$H_0 : f(F_j) = g(G_j) \quad \text{for all} \quad 1 \leq j \leq q.$$

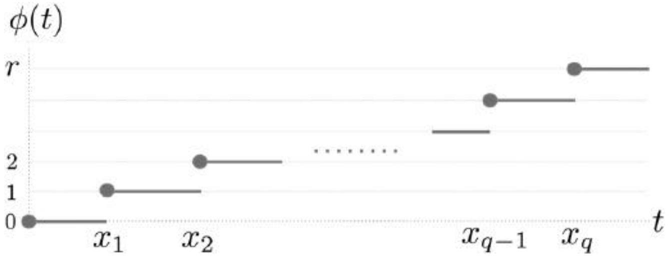


Figure 11.2 Monotonic features $x_1 \leq x_2 \leq \cdots \leq x_q$ are mapped to integers between 0 and q via some nondecreasing function $\phi(t)$.

For simplicity, we will simply assume there are q unique monotone features. We use

$$D_q = \sup_{1 \leq j \leq q} \left| f(F_j) - g(G_j) \right|$$

as a test statistic.

Under the null assumption, $f(F_j)$ and $g(G_j)$ are interchangeable. Thus, there are a total of $\binom{2q}{q}$ ways of permuting them. The distribution of D_q can be empirically determined using $\binom{2q}{q}$ permutations.

Theorem 11.2

$$P(D_q \geq d) = 1 - \frac{A_{q,q}}{\binom{2q}{q}},$$

where $A_{u,v}$ *satisfies* $A_{u,v} = A_{u-1,v} + A_{u,v-1}$ *with the boundary condition* $A_{0,q} = A_{q,0} = 1$ *within band* $|u - v| < d$.

Proof. The proof is similar to the combinatorial construction of Kolmogorov–Smirnov (KS) test (Böhm and Hornik, 2010; Gibbons and Chakraborti, 2011). Combine two monotonically increasing vectors

$$\big(f(F_1), \cdots, f(F_q)\big), \quad \big(g(G_1), \cdots, g(G_q)\big)$$

and arrange them in increasing order. Represent $f(F_j)$ and $g(G_j)$ as $\uparrow$ and $\rightarrow$ respectively. For example, $\uparrow\uparrow\rightarrow\uparrow\rightarrow\rightarrow \cdots$. There are exactly q number of $\uparrow$ and q number of $\rightarrow$ in the sequence. Treat the sequence as walks on a Cartesian grid. $\rightarrow$ indicates one step to the right and $\uparrow$ indicates one step up. The observed integer values of $(f(F_j), g(G_k))$ correspond to the coordinates of the grid. Thus the walk starts at $(0, 0)$ and ends at (q, q) (Figure 11.3). There are a total of $\binom{2q}{q}$ possible number of paths, which forms the sample space. This is also the total number of possible permutations between the elements

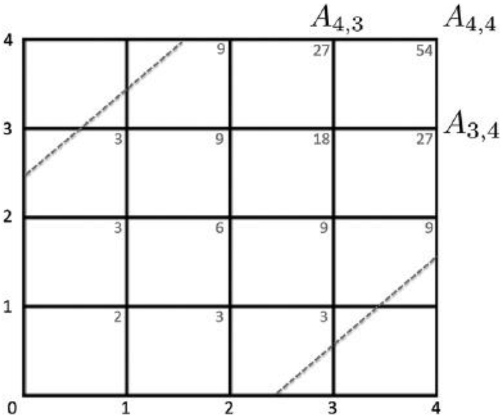


Figure 11.3 $A_{u,v}$ are computed within the boundary (dotted lines). The gray numbers are the number of paths from $(0,0)$; here there are 54 possible paths within the dotted lines from $(0,0)$ to $(4,4)$.

of the two vectors. The null assumption is that the two vectors are identical and there is no preference to one vector element to another. Thus, each walk is equally likely to happen in the sample space. Subsequently the probability can be written as

$$P(D_q \geq d) = 1 - P(D_q < d) = 1 - \frac{A_{q,q}}{\binom{2q}{q}}, \tag{11.2}$$

where $A_{u,v}$ is the total number of passible paths from $(0,0)$ to (u,v) within the boundary. Since there are only two paths (either ↑ or →), $A_{u,v}$ can be computed recursively as

$$A_{u,v} = A_{u-1,v} + A_{u,v-1}$$

within the boundary. On the boundary, $A_{0,q} = A_{q,0} = 1$ since there is only one path. □

Example 11.2 *Theorem 11.2 provides the exact probability computation for any number of nodes p. For instance, probability $P(D \geq 2.5)$ is computed iteratively as follows. We start with computing*

$$\begin{aligned} A_{1,1} &= A_{0,1} + A_{1,0} = 2, \\ A_{2,1} &= A_{1,1} + A_{1,0} = 3, \\ &\cdots, \\ A_{4,4} &= A_{4,3} + A_{3,4} = 27 + 27 = 54. \end{aligned}$$

These are given in Figure 11.3. Thus the probability (p-value) is computed as

$$P(D \geq 2.5) = 1 - 54 \Big/ \binom{8}{4} = 0.23.$$

A few other examples that can be computed easily are

$$P(D = 0) = 0, \; P(D \geq 1) = 1,$$
$$P(D \geq q) = 2 \Big/ \binom{2q}{q}, P(D \geq q+1) = 0.$$

Other examples that can be computed easily are

$$P(D_q = 0) = 0, \; P(D_q \geq 1) = 1$$

$$P(D_q \geq q) = 2 \Big/ \binom{2q}{q}, \; P(D_q \geq q+1) = 0.$$

Computing $A_{q,q}$ iteratively requires at most q^2 operations while permuting two samples consisting of q elements each requiring $\binom{2q}{q}$ operations. Thus, our method can compute the p-value exactly substantially faster than the permutation test that is approximate and exponentially slow. MATLAB code `PH_A.m` for computing $A_{q-1,q-1}$ is available.[1]

The asymptotic probability distribution of D_q can be also determined for sufficiently large q without computing iteratively as Theorem 11.2.

Theorem 11.3 $\lim_{q\to\infty} P\left(D_q/\sqrt{2q} \geq d\right) = 2\sum_{i=1}^{\infty}(-1)^{i-1}e^{-2i^2d^2}$.

The proof follows from Gibbons and Chakraborti (2011) and Smirnov (1939). From Theorem 11.3, p-value under H_0 is computed as

$$p\text{-value} = 2e^{-d_o^2} - 2e^{-8d_o^2} + 2e^{-18d_o^2} \cdots,$$

where d_o is the *least integer greater than* $D_q/\sqrt{2q}$ in the data (Figure 11.4). For any large observed value $d_0 \geq 2$, the second term is in the order of 10^{-14} and insignificant. Even for small observed d_0, the expansion converges quickly and five terms are sufficient.

11.2.1 Inference on Minimum Spanning Trees

In brain imaging, minimum spanning trees (MSTs) are often used in speeding up computation and simplifying complex graphs as simpler trees (Wu and Leahy, 1993). Wu et al. used MST in edge-based segmentation of a lesion

[1] http://www.stat.wisc.edu/~mchung/twins/

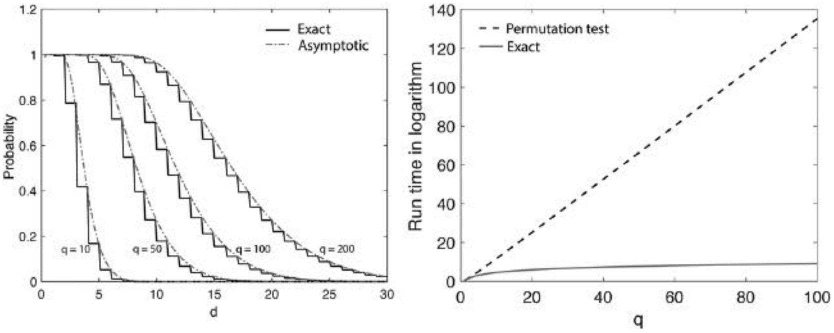


Figure 11.4 Left: convergence of Theorem 11.2 to Theorem 11.3 for $q = 10, 50, 100, 200$. Right: Run time of permutation test (dotted line) vs. the combinational method in Theorem 11.2 (solid line) in logarithmic scale.

in a brain MRI (Wu and Leahy, 1993). Stam et al. used MST as an unbiased skeleton representation of complex brain networks (Stam et al., 2014).

Most statistical inference methods on MST rely on existing univariate statistical test procedures on scalar graph theory features on MST such as the average path length. Since the probability distribution of such features is often not well known, resampling techniques such as permutation tests are frequently used. Permutation testing is known as the only exact test procedure. However, it is not exact in practice and only an approximate method due to the computational bottleneck of generating every possible permutation, which can be astrologically large. Thus, the statistical significance is computed using the small subset of all possible permutations, which gives approximate p-values. Here, we present a new combinatorial inference approach to the permutation test, where every possible permutation is enumerated combinatorially.

The MST is often constructed using Kruskal's algorithm (Lee et al., 2012). Kruskal's algorithm is a greedy algorithm with run time $O(p \log p)$ that starts with an edge with the smallest weight. Then add an edge with the next smallest weight. This sequential process continues while avoiding a loop and generates a spanning tree with the smallest total edge weights (Figure 11.5). Thus, the edge weights in the MST correspond to the order, in which the edges are added in the construction of the MST. Since there are p nodes, there are $p - 1$ edge weights in the MST.

Let M_1 and M_2 be the MST corresponding to $p \times p$ connectivity matrices C_1 and C_2. We are interested in testing hypotheses

$$H_0 : M_1 = M_2 \text{ vs. } H_1 : M_1 \neq M_2. \tag{11.3}$$

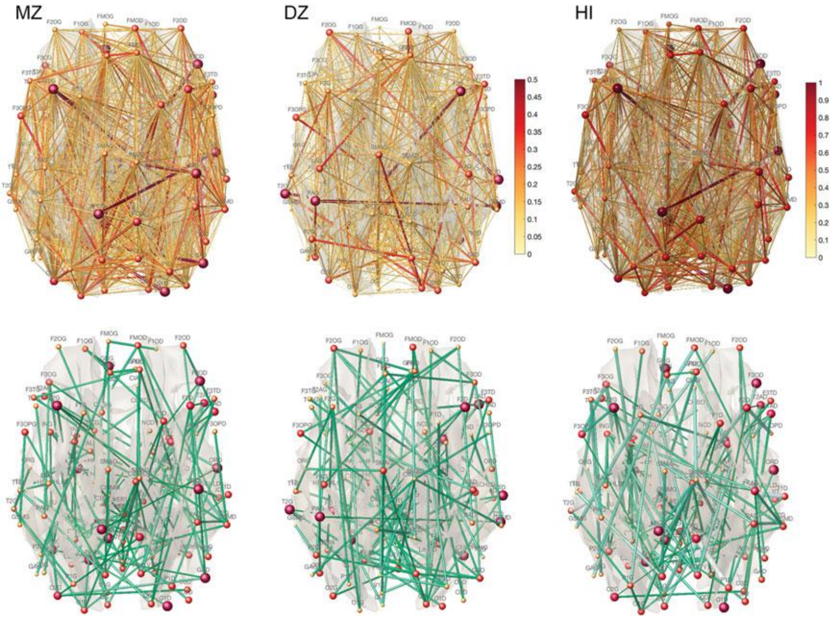


Figure 11.5 Top: Spearman correlation network of MZ- and DZ-twins and heritability index (HI) obtained from white matter fiber tract counts in DTI (Chung et al., 2018b). Bottom: corresponding MST on 1-correlation and HI constructed using Kruskal's algorithm.

Let

$$w_1^1 < w_2^1 < \cdots < w_{p-1}^1$$
$$w_1^2 < w_2^2 < \cdots < w_{p-1}^2$$

be the ordered edge weights of two MSTs in Kruskal's algorithm. w_j^1 and w_j^2 are edge weights obtained in the jth iteration of Kruskal's algorithm. Let f, g be monotone functions that map the edge weight obtained in the jth iteration to integer j, i.e.,

$$f(w_j^1) = j, \quad g(w_j^2) = j.$$

Figure 11.6 displays monotone functions f and g for an example with four nodes.

Under H_0, the sequence of monotone functions $f(w_j^1)$ and $g(w_j^2)$ are identical and interchangeable. The pseudometric

$$D(M_1, M_2) = \max_t \left| f(t) - g(t) \right|$$

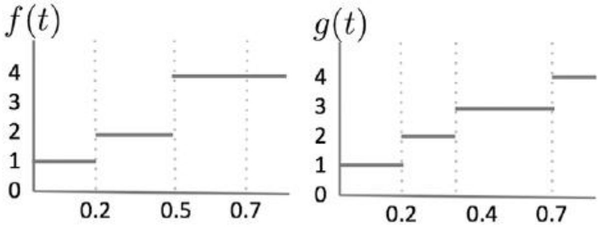


Figure 11.6 The number of nodes (vertical) in a subtree of an MST obtained in Figure 3.1. The horizontal axis is the edge weights where nodes are connected to the subtrees.

is used as the test statistic. Under H_0, $D(M_1, M_2) = 0$. The larger the value of D, it is more likely to reject H_0. Then we have shown that the probability distribution of D can be written as

$$P(D \geq d) = 1 - P(D < d) = 1 - \frac{A_{p-1,p-1}}{\binom{2p-2}{p-1}}, \tag{11.4}$$

where $A_{u,v}$ is the total number of passible paths from $(0,0)$ to (u, v) within the boundary (Chung et al., 2017a).

Figure 11.7 displays monotone functions $f(t)$ and $g(t)$ obtained from an MST of MS- and DS-twins. At edge weight 0.75, which is the maximum gap and corresponding to correlation 0.25, the observed distance D was 46. The corresponding p-value was computed as $P(D \geq 46) = 1.57 \times 10^{-8}$. The localized regions of brain that genetically contribute the most can also be identified by identifying the nodes of connections around edge weight 0.75 (0.75 ± 0.2). The following AAL regions are identified as the region of statistically significant MST differences: Frontal-Mid-L, Frontal-Mid-R, Frontal-Inf-Oper-R, Rolandic-Oper-R, Olfactory-L, Frontal-Sup-Medial-L, Frontal-Sup-Medial-R, Occipital-Inf-L, SupraMarginal-R, Precuneus-R, Caudate-L, Putamen-L, Temporal-Pole-Sup-L, Temporal-Pole-Sup-R, Temporal-Pole-Mid-R, Cerebelum-Crus2-R, Cerebelum-8-R, and Vermis-8. The identified frontal and temporal regions are overlapping with the previous MRI-based twin study (Thompson et al., 2001).

11.2.2 Application to Multiple Subjects

The exact combinatorial inference is designed to test the equivalence of two networks. If we have multiple networks, the method is not directly applicable and some modification is needed.

Suppose group I consists of m connectivity matrices $C_1^1, C_2^1, \cdots, C_m^1$. Suppose group II consists of n connectivity matrices $C_1^2, C_2^2, \cdots, C_n^2$. We are

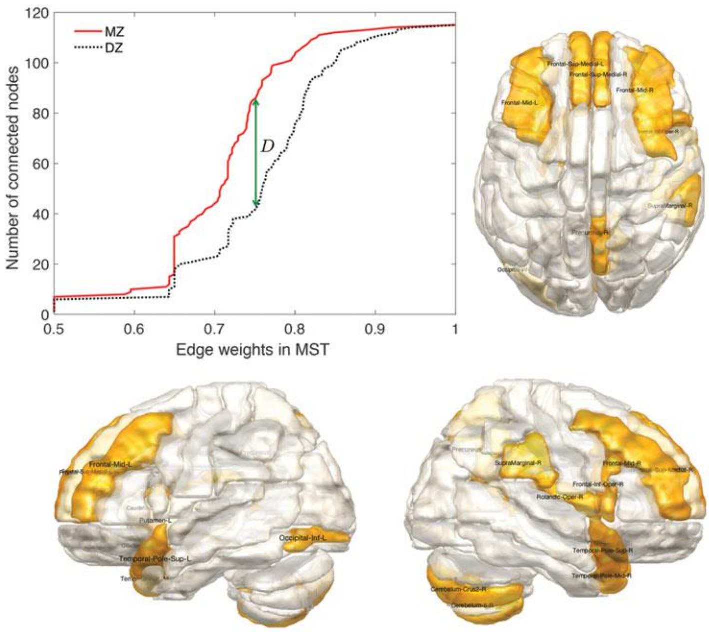


Figure 11.7 The number of connected nodes is plotted over the edge weights of the MST obtained from the Spearman correlation matrices of MZ- (solid red) and DZ-twins (dotted black). The KS-distance statistic D is 45 at edge weight 0.75. Regions correspond to the maximum difference between MSTs of MZ- and DZ-twins. They correspond to regions that are in $[0.55, 0.95]$ range in distance.

interested in testing the equivalence of connectivity matrices between groups I and II. The exact combinatorial inference cannot be applicable directly. Consider the entries of connectivity matrices

$$C_i^j = (c_i^j(k,l)).$$

In this notation, the (k,l)th entry of connectivity matrix C_i^j is $C_i^j(k,l)$. There are two different ways to apply the exact combinatorial inference. In the first approach, we compute the similarity or distance between all the (k,l)th entries of the connectivity matrix. For instance, we correlate all the (k,l)th entries in groups I and II separately:

$$\begin{aligned} a_{kl} &= corr(C_1^1(k,l), C_2^1(k,l), \cdots, C_m^1(k,l)), \\ b_{kl} &= corr(C_1^2(k,l), C_2^2(k,l), \cdots, C_n^2(k,l)). \end{aligned}$$

Then we obtain two new group-level connectivity matrices $A = (a_{kl})$ and $B = (b_{kl})$. Now we can apply the exact combinatorial inference testing the difference between A and B.

In the first approach, we computed the distance of all the subjects simultaneously at each fixed (k,l)th entry. In the second approach, we compute the pairwise distance between two connectivity matrices for all the subjects. We can use GH- or KS-distance. Let $d_{ij}^1 = d(C_i^1, C_j^1)$ be the distance between connectivity matrices C_i^1 and C_j^1. Similarly, we define d_{ij}^2 be the distance betewen C_i^2 and C_j^2. Then we obtain two group-level connectivity matrices $D^1 = (d_{ij}^1)$ and $D^2 = (d_{ij}^2)$. Now we can perform the exact combinatorial inference.

11.2.3 Inference on the Number of Cycles

Given two graphs G_1 and G_2, we will use the difference of their β_1 in differentiating the graphs. Consider distance

$$D(G_1, G_2) = \sup_\lambda \big|\beta_1(G_1|_\lambda) - \beta_1(G_2|_\lambda)\big|,$$

where $G_i|_\lambda$ denotes the binary network thresholded below at edge weight λ. Note

$$\beta_1(G_i|_\lambda) = l_i(\lambda) - p + \beta_0^i(\lambda),$$

where p is the number of nodes of G_i, which is fixed; $l_i(\lambda)$ is the number of edges of $G_i|_\lambda$; and $\beta_0^i(\lambda)$ is the number of connected components of $G_i|_\lambda$. Then the distance can be written as

$$D(G_1, G_2) = \sup_\lambda \big|l_1(\lambda) - l_2(\lambda) + \beta_0^1(\lambda) - \beta_0^2(\lambda)\big|.$$

Under the null hypothesis $H_0 : G_1 = G_2$, we may assume the number of edges are identical. Then, the p-value is bounded by

$$P(D(G_1, G_2) > d) \leq P\Big(\sup_\lambda \big|\beta_0^1(\lambda) - \beta_0^2(\lambda)\big| > d\Big).$$

This is the probability distribution of KS-distance introduced in Chung et al. (2017a).

11.2.4 Validation of Exact Combinatorial Inference

In most of brain imaging studies, 5,000 to 100,000 permutations are often used, which puts the total number of permutations usually less than 3% of all possible permutations. In Zalesky et al. (2010), 5,000 permutations out of a possible $\binom{27}{12} = 17,383,860$ permutations (2.9%) were used. In Thompson et al. (2001), 1 million permutations out of $\binom{40}{20}$ possible permutations (0.07 %) were used.

For validation and comparisons, we simulated the random graphs with the ground truth. We used $p = 40$ nodes and $n = 10$ images, which makes possible permutations to be exactly $\binom{10+10}{10} = 184,756$, making the permutation test manageable. The data matrix $X_{n \times p} = (x_{ij}) = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p)$ is simulated as standard normal in each component, i.e., $x_{ij} \sim N(0,1)$, or equivalently each column is multivariate normal $\mathbf{x}_j \sim N(0, I)$ with the identity matrix as the covariance.

Let $Y = (y_{ij}) = (\mathbf{y}_1, \cdots, \mathbf{y}_p) = X$. So far, there is no statistical dependency between nodes in Y. We add the following block modular structure to Y. We assume there are $k = 4, 5, 8, 10, 40$ modules and each module consists of $c = p/k = 10, 8, 5, 4, 1$ number of nodes. Then for the ith node in the jth module, we simulate

$$\mathbf{y}_{c(j-1)+i} = \mathbf{x}_{c(j-1)+1} + N(0, \sigma I) \qquad \text{for } 1 \le i \le c, 1 \le j \le k \tag{11.5}$$

with $\sigma = 0.1$. Subsequently, the connectivity matrix $C = (c_{ij})$ is given by $c_{ij} = corr(\mathbf{y}_i, \mathbf{y}_j)$. This introduces the block modular structure in the correlation network (Figure 11.8). For 40 modules, each module consists of just one node, which is basically a network with zero modules.

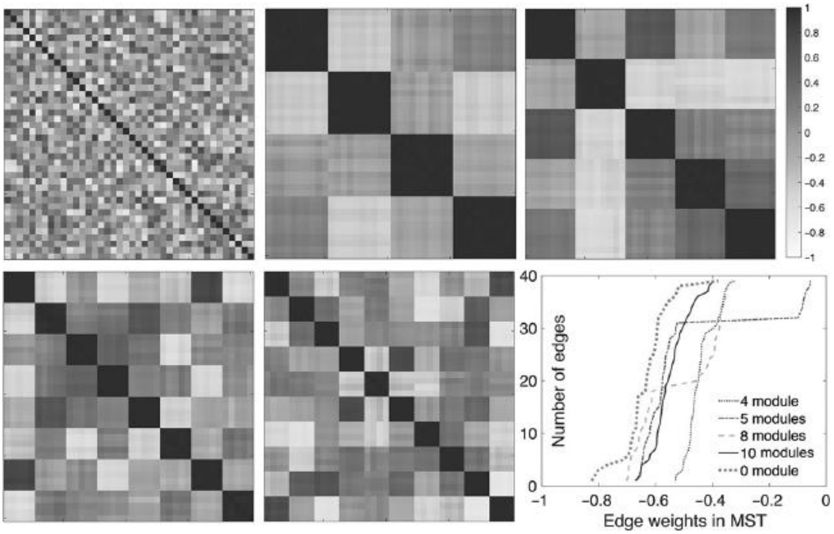


Figure 11.8 Randomly simulated correlation matrices with zero, four, five, eight, and 10 modules. The plot shows the number of nodes over the largest edge weights added into MST construction during Kruskal's algorithm for four, five, eight, 10, and zero modules.

Table 11.1. *Simulation results given in terms of p-values. In the case of no network differences (0 vs. 0 and 4 vs. 4), higher p-values are better. In the case of network differences (4 vs. 5, 4 vs. 8, and 5 vs. 10), smaller p-values are better.*

	Combinatorial	Permute 0.1%	Permute 0.5%	Permute 1%
0 vs. 0	0.831 ± 0.187	0.746 ± 0.196	0.745 ± 0.195	0.744 ± 0.196
4 vs. 4	0.456 ± 0.321	0.958 ± 0.075	0.958 ± 0.073	0.958 ± 0.073
4 vs. 5	0.038 ± 0.126	0.381 ± 0.311	0.377 ± 0.311	0.378 ± 0.311
4 vs. 8	0.053 ± 0.138	0.410 ± 0.309	0.411 ± 0.306	0.411 ± 0.306
5 vs. 10	0.060 ± 0.126	0.391 ± 0.283	0.395 ± 0.284	0.395 ± 0.283

Using (11.5), we simulated random networks with four, five, eight, 10, and zero modules. For each network, we obtained an MST and computed the distance D between networks. We computed the p-value using the combinatorial method. In comparison, we performed the permutation tests by permuting the group labels and generating 0.1, 0.5, and 1% of every possible permutation. The procedures are repeated 100 times and the average results are reported in Table 11.1.

In the case of no network differences (zero vs. zero and four vs. four), higher p-values are better. The combinatorial method and the permutation tests all performed well for no network difference. In the case of network differences (four vs. five, four vs. eight, and five vs. 10), smaller p-values are better. The combinatorial method performed far superiorly than the permutation tests. None of the permutation tests detected modular structure differences. The proposed combinatorial approach on an MST seems to be far more sensitive in detecting modular structures. The performance of the permutation test does not improve even when we sample 10% of all possible permutations. The permutation test doesn't converge rapidly with increased samples. The codes for performing exact combinatorial inference as well as simulations is provided.[2]

11.3 Bootstrap

Although not as popular as the permutation test, the bootstrap can be very effective in estimating network parameters. The *bootstrap* requires no parametric assumptions in doing statistical inference (Efron, 1982). Given a

[2] http://www.stat.wisc.edu/~mchung/twins/

random sample $\mathbf{x} = (x_1, \cdots, x_n)$, the cumulative distribution function of true population F is empirically estimated as the proportion of the total sample points that is less than x, i.e.,

$$\widehat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{I}_{(-\infty, x_i)}(x),$$

where $\mathcal{I}_A(x)$ is an index function taking value 1 if $x \in A$ and 0 otherwise. For order statistics $x_{(1)}, \cdots, x_{(n)}$ of $\mathbf{x}$, we have

$$\widehat{F}(x_{(1)}) = \frac{1}{n}, \cdots, \widehat{F}(x_{(i)}) = \frac{i}{n} \text{ and } \widehat{F}(x_{(i+1)}) = \frac{i+1}{n}.$$

Empirically, we can assume each sample point to occur with equal probability $1/n$. Then we *resample with replacement* from $\mathbf{x}$ with equal probability. The new resample is denoted as $\mathbf{x}^*$ and called a *bootstrap sample*. If we resample m times, we denote them as $\mathbf{x}^{*1}, \cdots, \mathbf{x}^{*m}$. The MATLAB code for doing resampling with replacement three times for $\mathbf{x} = (5, 8, 3, 2)$ is as follows:

```
>boot=inline('x(unidrnd(length(x),m,length(x)))',...
'x','m')
boot =
     Inline function:
     boot(x,m) = x(unidrnd( length(x), m,length(x)))


>x=[5 8 3 2];
>bs=boot(x,3)
bs = 2     3     8     8
     3     8     3     2
     2     8     3     3
```

Suppose sample $\mathbf{x} = (x_1, \cdots, x_n)$ follows distribution $F(\theta)$ for some parameter θ. We are interested in performing an inference on θ. For this, we need to know the distribution $\widehat{\theta}(\mathbf{x})$, which is the estimate of θ. Unfortunately, the distribution may be difficult to obtain analytically. Thus, we generate m bootstrap samples $\mathbf{x}^{*1}, \cdots, \mathbf{x}^{*m}$ of $\mathbf{x}$ first. Then obtain the ith bootstrap replication of $\widehat{\theta}$ as

$$\widehat{\theta}^{*i} = \widehat{\theta}(\mathbf{x}^{*i}).$$

Instead of using the original sample $\mathbf{x}$ in estimating θ, we use the ith bootstrap replication $\mathbf{x}^{*i}$ in estimating θ and denote it as $\widehat{\theta}^{*i}$. These bootstrap resamples provide us with an estimate of the distribution of $\widehat{\theta}$. In particular, we can estimate mean and the variance of θ as

$$\mathbb{E}\,\widehat{\theta}(\mathbf{x}) \approx \bar{\theta}^* = \frac{1}{m}\sum_{i=1}^{m}\widehat{\theta}^{*i},$$

$$\mathbb{E}\,\widehat{\theta}(\mathbf{x}) \approx \frac{1}{m-1}\sum_{i=1}^{m}\left(\widehat{\theta}^{*i} - \bar{\theta}^*\right)^2.$$

The following sample data published in Blakley et al. (1994) can be used as an illustration for this section. The data can be downloaded from the web.[3] There are two measurements, `arm` and `grip`, that measure the arm and grip strength of construction workers. From the data, let us estimate the population mean and variance based on 200 bootstrap replications. For estimating the population mean and variance, we use $\widehat{\theta}(\mathbf{x}) = \bar{x}$ and $\widehat{\theta}(\mathbf{x}) = n\bar{x}$.

```
bs=boot(arm,200);
bsmean=mean(bs,2)
>>[mean(arm) mean(bsmean)]
 ans =
   78.7517   78.9025
```

The built-in MATLAB function `bootstrp`, which is fairly limited in its functionality, can be also used:

```
>mean(bootstrp(200,'mean',arm))
ans =
   78.7799
```

The performance of an estimator is determined by bias. Ideally we like to have an unbiased estimator. $\mathbb{V}\widehat{\theta}(\mathbf{x})$ measures the performance of a bootstrap estimate. To measure the accuracy of an estimate, we compute the *bias*, which is defined as the difference between the expection of the estimator and the true parameter,

$$bias(\widehat{\theta}) = \mathbb{E}\widehat{\theta} - \theta.$$

The bootstrap estimation of the bias is then given by

$$bias(\bar{\theta}^*) = \bar{\theta}^* - \widehat{\theta}.$$

For our example, for estimating the population mean, let $\widehat{\theta} = \bar{X}$:

$$bias(\bar{X}^*) = \bar{X}^* - \bar{X}.$$

[3] http://www.stat.wisc.edu/~mchung/teaching/data/strength.data

Based on the MATLAB result, we have

```
>>78.9025-78.7517
0.1508.
```

The bootstrap estimation is not showing severe bias relative to the size of the signal.

11.3.1 Bootstrap Confidence Intervals

Suppose random variable $X \sim \frac{1}{\sigma} f(\frac{x-\mu}{\sigma})$ for some distribution f. We want to find $100(1-\alpha)\%$ confidence interval (CI) for estimated parameter $\widehat{\mu} = \bar{X}$. We need to find a pivot of μ. A pivot is a statistic that contains μ but whose distribution does not depend on μ or σ. For example, $f \sim N(\mu, \sigma^2)$,

$$T = \frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t_{n-1}$$

is a pivot for μ. Let $t_{\alpha/2}$ and $t_{1-\alpha/2}$ be $\alpha/2$ and $1-\alpha/2$ quantiles for t_{n-1} distribution. The $100(1-\alpha)\%$ CI is given by

$$\Big(\bar{x} - t_{1-\alpha/2} S/\sqrt{n},\ \bar{x} - t_{\alpha/2} S/\sqrt{n}\Big).$$

If f is not normal, T will be still a pivot, but we do not know its distribution. So we generate 1,000 bootstrap replicates $T^* = \frac{\bar{X}^* - \bar{X}}{S^*/\sqrt{n}}$ of T and get the bootstap CI given by

$$\Big(\bar{x} - t^*_{1-\alpha/2} S/\sqrt{n},\ \bar{x} - t^*_{\alpha/2} S/\sqrt{n}\Big).$$

This is computed as follows in MATLAB:

```
t=inline('(mean(x)-u)/(std(x)/length(x))','x','u')
for i=1:1000;
  trep(i)=t(boot(arm,1),mean(arm))
end;
> quantile(trep,[0.05 0.95])
  -19.8246   22.0677
> mean(arm) -22.0677*std(arm)/sqrt(147)
   40.3303
> mean(arm) +19.8246*std(arm)/sqrt(147)
  113.2677
```

Thus the 90% bootstrap confidence interval is $[40.33, 113.27]$.

11.3.2 Bootstrap Estimation in Linear Models

For given bivariate data $(x_i, y_i), i = 1, \cdots, n$, suppose we have the following regression model:

$$Y_i = g(x_i; \beta) + \epsilon_i.$$

For instance, consider following linear model $g(x;\beta) = \beta_0 + \beta_1 x$, where $\beta = (\beta_0, \beta_1)$. Fitting a nonlinear model like $g(x;\beta) = \beta_0/(\beta_1 - \beta_2 e^{-\beta_3 x})$ is also possible. The least squares estimation (LSE) of β is given by

$$\widehat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} \Big(y_i - g(x_i;\beta)\Big)^2.$$

Based on LSE, the observed residual is

$$\widehat{\epsilon_i} = y_i - g(x_i; \widehat{\beta}). \tag{11.6}$$

Then we resample the residual. A bootstrap replicate is then obtained as

$$y_i^* = g(x_i; \widehat{\beta}) + \widehat{\epsilon}_i^*. \tag{11.7}$$

The bootstrap estimate of β is then

$$\widehat{\beta}^* = \arg\min_{\beta} \sum_{i=1}^{n} \Big(y_i^* - g(x_i;\beta)\Big)^2. \tag{11.8}$$

Consider estimating the CI of β_1 for model

$$\texttt{grip} = \beta_0 + \beta_1 \texttt{arm}$$

in the strength data. The pivot for β_1 is $T = \frac{\widehat{\beta}_1 - \beta_1}{SE(\widehat{\beta}_1)}$, where $SE(\widehat{\beta}_1) = \sqrt{\mathbb{V}\widehat{\beta}_1}$. From equations (11.6), (11.7), and (11.8), we generate 1,000 bootstrap replicates of $T^* = \frac{\widehat{\beta}_1^* - \widehat{\beta}}{SE\widehat{\beta}_1^*}$. Since we don't know the standard error $SE(\widehat{\beta}_1)$, we need to estimate $SE(\widehat{\beta}_1^*)$ via bootstrap, which requires further bootstrapping from bootstrapped residual e_i^*. Let us denote this estimate as $S_{\widehat{\beta}_1}$. Then $100(1-\alpha)\%$ CI for β_1 is

$$(\widehat{\beta}_1 - t_{1-\alpha/2}^* S_{\widehat{\beta}_1},\ \widehat{\beta}_1 - t_{\alpha/2}^* S_{\widehat{\beta}_1}).$$

In MATLAB, this is computed as follows:

```
beta=inline('pinv([ones(147,1) x])*y')
>b=beta(arm,grip)
b=
   54.7081
   0.7050        % least squares estimation of beta1.

e=grip-b(1)-b(2)*arm;      %residuals
```

```
for j=1:1000
  bse=boot(e,1);            %bootstrap on residuals
  bsgrip = bse + b(1) + b(2)*arm;
  bsb=beta(arm,bsgrip);

  for i=1:50          %estimating the s.d. of beta1.
     bs2e=boot(bse,1);     %bootstrap on bootstrap
     bs2grip = bs2e + b(1) + b(2)*arm;
     bs2b=beta(arm,bs2grip);
     bsbeta(i)=bs2b(2);
  end;
  trep(j)=(bsb(2)-0.7050)/std(bsbeta);
end;

>quantile(trep,[0.05 0.95])
-1.5308    1.6580
```

12

Series Expansion of Connectivity Matrices

In this chapter, we explain how to expand connectivity matrices as a series expansion for subsequent statistical inference. One of the simplest well-known expansions is the spectral decomposition, which is the basis of the principal component analysis (PCA). However, there are many other possible expansions as well.

12.1 Spectral Decomposition

Often brain connectivity matrices are symmetric. The connectivity matrices may not be positive definite if the number of images n is smaller than the number of nodes p, which falls into the *small-n large-p problem*. We will consider general connectivity matrices and we will not assume positive definiteness.

Suppose we have a $p \times p$ symmetric matrix C with eigenvalues

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_p$$

and the corresponding eigenvectors $v_1, v_2, \cdots, v_p$ such that

$$Cv_j = \lambda_j v_j.$$

Large eigenvalues correspond to the high-frequency component. We will *not* assume the positive definiteness of C, and we may expect negative eigenvalues as well (Figure 12.1). If C is positive definite, all the eigenvalues are positive. If C is nonnegative definite, some eigenvalues are zeros.

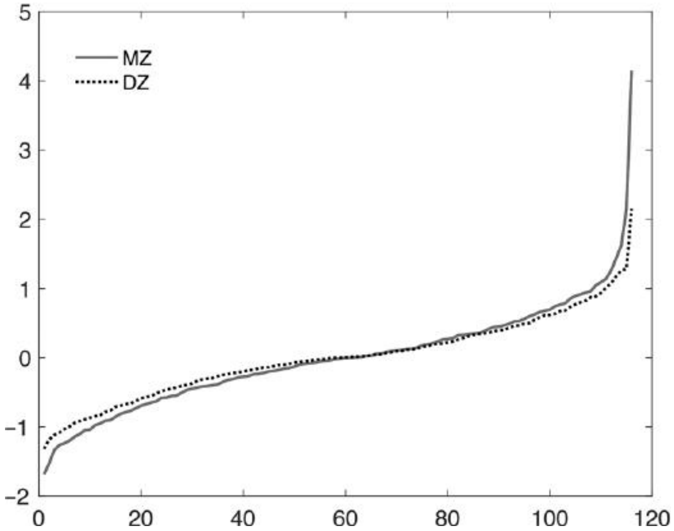


Figure 12.1 Eigenvalues of Spearman correlation matrices from MZ- (solid line) and DZ-twins (dotted line) obtained from a DTI study (Chung et al., 2018b). We have negative eigenvalues due to the small sample size n relative to the number of nodes p. Also DTI produces sparse connectivity matrices, which reduces the rank of matrices and in turn produces negative eigenvalues. Higher eigenvalues correspond to the high-frequency signal content.

We further assume $v_1, v_2, \cdots, v_p$ are orthonormal. Let $Q = [v_1 v_2 \cdots v_p]$ be an orthogonal matrix such that $Q'Q = I$, the identity matrix. Then we have

$$C[v_1 v_2 \cdots v_p] = CQ = [v_1 v_2 \cdots v_p]D = QD$$

with $D = diag(\lambda_1, \lambda_2, \cdots, \lambda_p)$. Then trivially we have the decomposition:

$$C = QDQ^\top$$

An algebraic manipulation can show that C can be written as

$$C = \sum_{j=1}^{p} \lambda_j v_j v_j^\top. \tag{12.1}$$

Equation (12.1) is often known as the spectral decomposition of C. Since correlation and covariance matrices are often symmetric and positive definite, they can be decomposed in this fashion. However, any symmetric matrix can be

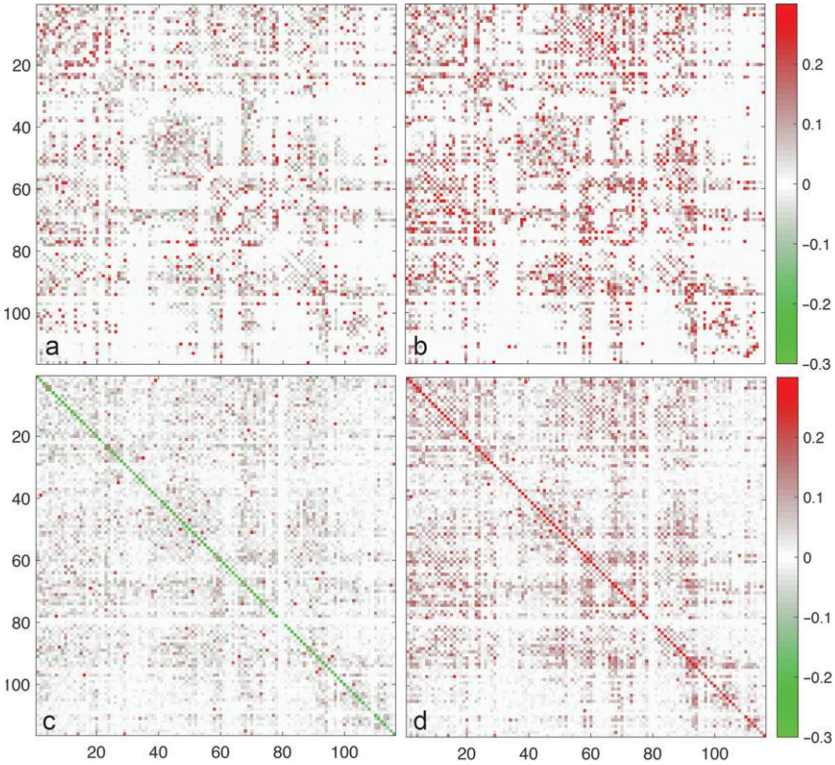


Figure 12.2 (a) DZ-twin correlation. Each entry is Spearman correlation of the number of tracts between 116 regions in the AAL parcellation. (b) MZ-twin correlation. (c) Spectral decomposition with negative eigenvalues. (d) Spectral decomposition with positive eigenvalues. Correlation matrix B is equal to the sum of negative and positive decompositions.

decomposed in this fashion. In fact, we can decompose a connectivity matrix into negative and positive eigenvalue parts (Figure 12.2):

$$C = \sum_{\lambda_j <= 0} \lambda_j v_j v_j^\top + \sum_{\lambda_j > 0} \lambda_j v_j v_j^\top .$$

12.2 Iterative Residual Fitting

The iterative residual fitting (IRF) algorithm is an iterative procedure for solving large-scale computational problems that cannot be solved with existing computational recourses into a smaller problems. Sequentially combining the solutions of the subproblems, we obtain the convergence to the larger problem. The method is first introduced in Chung et al. (2008a) in solving the large-scale spherical harmonic expansion problem for brain surfaces. IRF can be used to expand connectivity matrices with respect to other matrices.

12.2.1 Computational Issues of Fourier Series Expansion

Consider a manifold $\mathcal{M} \in \mathbb{R}^d$ that will be our object of interest. Let $L^2(\mathcal{M})$ be the space of square integrable functions in $\mathcal{M}$ with inner product

$$\langle g_1, g_2 \rangle = \int_{\mathcal{M}} g_1(p) g_2(p) \, d\mu(p), \tag{12.2}$$

where μ is some measure such that $\mu(\mathcal{M})$ is the total volume of $\mathcal{M}$. The norm $\| \cdot \|$ is defined as

$$\|g\| = \langle g, g \rangle^{1/2}.$$

The operator $\mathcal{L}$ is *self-adjoint* if

$$\langle g_1, \mathcal{L} g_2 \rangle = \langle \mathcal{L} g_1, g_2 \rangle$$

for all $g_1, g_2 \in L^2(\mathcal{M})$. The eigenvalues λ_j and eigenfunctions ψ_j of the operator $\mathcal{L}$ are obtained by solving

$$\mathcal{L} \psi_j = \lambda_j \psi_j. \tag{12.3}$$

Without the loss of generality, we can order eigenvalues

$$0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \cdots$$

and make the eigenfunctions to be orthonormal with respect to the inner product (12.2).

Let $\mathcal{H}_k$ be the subspace

$$\mathcal{H}_k = \left\{ \sum_{j=0}^{k} \beta_j \psi_j(p) : \beta_j \in \mathbb{R} \right\} \subset L^2(\mathcal{M}),$$

which is spanned by the finite number of basis up to degree k. We are interested in finding a function $h \in \mathcal{H}_k$ in the subspace that is the closest to an arbitrary function f in L_2-norm. From the property of Hilbert space $L^2(\mathcal{M})$, we have

$$\sum_{j=0}^{k} f_j \psi(p) = \arg\min_{h \in \mathcal{H}} \|f - h\|^2, \tag{12.4}$$

where $f_j = \langle f, \psi_j \rangle$ are Fourier coefficients. For complicated manifold $\mathcal{M}$ and large $k > 5,000$, the problem can cause a serious computational bottleneck. There are three well-known methods for computing Fourier coefficients.

The first method numerically integrates the Fourier coefficients by discretizing the problem (Chung, 2006). Although this approach is the simplest to implement numerically and possibly the most accurate, the computation can be extremely slow, due to the brute force nature of the technique. However, it is possible to speed up the computational drastically if the finite element method (FEM) is used and the inner product operations can be simplified as matrix multiplication (Chung et al., 2015b).

The second method is perhaps the most often used method and based on the fast Fourier transform (FFT) (Bulow, 2004; Gu et al., 2004). The drawback of FFT is the need for a predefined regular grid system so it is usually not applicable to more general manifold settings such as graphs and surface meshes. In particular, for brain surfaces, since the mesh topology is different for different subjects, a time-consuming interpolation or registration is needed to align all the subject to the standard regular grid system. Note cortical meshes obtained from FreeSurfer (Fischl and Dale, 2000) produces topologically different meshes for different subjects, so FFT is not applicable in such situations.

The third method is based on solving a system of linear equations (Gerig et al., 2001; Shen et al., 2004; Shen and Chung, 2006) in a least squares fashion, which requires matrix inversion. This is the most widely used numerical technique in the spherical harmonics literature. However, the direct application of the least squares estimation is not desirable when the size of the linear equation is extremely large.

For extremely large least squares problems, new iterative strategies such as the IRF is required (Chung et al., 2008a). Suppose f is observed at the finite number of points $p_1, \cdots, p_n \in \mathcal{M}$. Then we wish to find $h \in \mathcal{H}_k$, which minimizes the sum of the squared distance

$$\|f - h\|^2 = \sum_{i=1}^{n} \left[f(p_i) - \sum_{j=0}^{k} \beta_j \psi_j(p_i) \right]^2 . \tag{12.5}$$

The minimum of (12.5) is obtained when

$$f(p_i) = \sum_{j=0}^{k} \beta_j \psi_j(p_i), i = 1, \cdots, n \tag{12.6}$$

Equation (12.6) is referred as the *normal equation* and is usually solved by matrix inversion.

Let $f = (f(p_1), \cdots, f(p_n))^\top$, $\beta = (\beta_0, \cdots, \beta_k)^\top$ and

$$\Psi = \begin{bmatrix} \psi_0(p_1) & \cdots & \psi_k(p_1) \\ \vdots & \ddots & \vdots \\ \psi_0(p_n) & \cdots & \psi_k(p_n) \end{bmatrix}$$

be a $n \times (k+1)$ matrix consisting of basis functions evaluated at mesh vertices. Then (12.6) can be rewritten in the following matrix form:

$$f = \Psi\beta. \tag{12.7}$$

The solution of the matrix equation is

$$\beta = (\Psi^\top \Psi)^- \Psi^\top f, \tag{12.8}$$

where $(\Psi^\top \Psi)^-$ is the generalized inverse. The problem with this widely used formulation is that the size of the matrix Ψ can be fairly large for very large n and k. So it is impractical to perform matrix operation (12.8) directly. This is mainly true for FreeSurfer (Fischl and Dale, 2000), which produces more than 300,000 nodes per each cortical hemisphere. This computational bottleneck can be overcome by breaking the least squares problem in the subspace $\mathcal{H}_k$ into smaller subspaces using the IRF algorithm (Chung et al., 2008a; Shen and Chung, 2006).

12.2.2 IRF Algorithm

We present an iterative technique for solving (12.7) for an extremely large number of bases k. Decompose the subspace $\mathcal{H}_k$ into smaller subspaces as the direct sum:

$$\mathcal{H}_k = \mathcal{I}_0 \oplus \mathcal{I}_1 \cdots \oplus \mathcal{I}_k,$$

where subspace $\mathcal{I}_l$ is the the projection of $\mathcal{H}_k$ along the kth basis. Another way of decomposing $\mathcal{H}_k$ is to use more than one basis for $\mathcal{I}_l$. For instance, for spherical harmonics Y_{lm} at degree l, there are $2l+1$ bases $Y_{l,-l}, \cdots, Y_{l,l}$. Thus, we can define $\mathcal{I}_l$ as the $2l+1$ dimensional subspace generated by all lth degree spherical harmonics. This has an advantage of simultaneously estimating more than one coefficient.

The algorithm estimates the Fourier coefficients β_j in each subspace $\mathcal{I}_j$ iteratively from increasing the degree from 0 to k. Suppose we estimated the coefficients up to degree $l-1$ somehow. The estimated coefficients are denoted as $\widehat{\beta_0}, \cdots, \widehat{\beta}_{l-1}$. Then the residual r_{l-1} of the fit is given by

$$r_{l-1} = f - \sum_{j=0}^{l-1} \widehat{\beta}_j \psi_j. \tag{12.9}$$

In practice, the computation is done in the finite number of discrete points p_1, $p_2, \cdots, p_n$. Thus $r_{l-1} = (r_{l-1}(p_1), \cdots, r_{l-1}(p_n))^\top$ is a residual vector in (12.9). At the next degree l, we estimate the coefficients β_l by minimizing the difference between the residual r_{l-1} and $\beta_l \psi_l$, i.e.,

$$\widehat{\beta_l} = \arg\min_{\beta_l} \|r_{l-1} - \beta_l \psi_l\|^2.$$

The minimization is achieved in the least squares fashion with a smaller normal equation. Let $\Psi_l = (\psi_l(p_1), \cdots, \psi_l(p_n))^\top$. Then the minimizer is given by

$$\begin{aligned}\widehat{\beta_l} &= (\Psi_l^\top \Psi_l)^{-1} \Psi_l^\top r_{l-1} \\ &= \frac{\Psi_l^\top r_{l_1}}{\sum_{i=0}^n \psi_j^2(p_i)}.\end{aligned}$$

The functional data f is then estimated by combining terms

$$\sum_{j=0}^{l} \widehat{\beta_j} \psi_j.$$

In this fashion, the algorithm hierarchically builds the Fourier expansion from lower to higher degree.

12.2.3 Speeding Up IRF

To speed up the computation, we can decompose $\mathcal{H}_k$ such that each subspace $\mathcal{I}_k$ is spanned by more than one basis if necessary. The iterative procedure presented here is referred to as the iterative residual fitting (IRF) algorithm since we are iteratively fitting a linear equation to the residuals obtained from the previous iteration (Chung et al., 2008a). However, one limitation with IRF is that it was known that the stepwise regression always underestimates the Fourier coefficients in absolute value.

The basis function ψ_j is orthonormal with respect to the integral version of the inner product. Hermite polynomials and spherical harmonics are such examples. If we reshape these analytic bases in such a way that they are orthonormal with respect to the vector product, i.e., $\Psi_i^\top \Psi_j = \delta_{ij}$, Kronecker's delta, we can drastically speed up the computation. Under this condition, the minimizer is simply given as

$$\widehat{\beta_l} = \Psi_l^\top r_{l_1}.$$

If the grid points $p_1, p_2, \cdots, p_n$ are dense enough, then the off-diagonal entries of $\Psi^\top \Psi$ are close to zeros, but the diagonal entries of $\Psi^\top \Psi$ are not going to be close to ones. Thus, simply modulating the diagonal entries, often we may achieve the desired effects. The best approach is to discretize the underlying manifolds as a graph with nodes $p_1, \cdots, p_n$ and compute the eigenvector of the graph Laplacian. The connectivity of the graph can be achieved by performing the abstract version of the Delaunday triangulation or Rips complex construction (Wang et al., 2017).

12.2.4 Best Model Selection

In many spherical harmonic representation literature (Bulow, 2004; Gerig et al., 2001; Gu et al., 2004; Shen and Chung, 2006; Shen et al., 2004), the optimal degree is simply selected based on a prespecified error bound that depends on the magnitude of estimating functions. Although increasing the degree of the representation increases the goodness-of-fit, it also increases the number of coefficients to be estimated quadratically. So it is necessary to find the optimal degree where the goodness-of-fit and the number of parameters balance out. The stepwise model selection framework offers a way to automatically determine the optimal degree (Chung et al., 2008a).

Suppose we can have expansion

$$f(p_i) = \sum_{j=0}^{k-1} \beta_j \psi_j(p_i) + \epsilon(p_i), \tag{12.10}$$

where $\epsilon(p_i)$ is a zero mean Gaussian random variable. Then we determine if adding the kth degree terms in the $(k-1)$th degree model (12.10) is statistically significant by testing the null hypothesis

$$H_0 : \mu_k = 0.$$

Let the kth degree sum of squared errors (SSE) be

$$\mathrm{SSE}_k = \sum_{i=1}^{n} r_k^2(p_i).$$

As the degree k increases, SSE keep decreasing until it flattens out. So it is reasonable to stop the iteration when the decrease in error is no longer significant. Figure 12.3 shows the plot of the RMSE, $\sqrt{\mathrm{SSE}_k/n}$. Under H_0, the test statistic is

$$F = \frac{\mathrm{SSE}_{k-1} - \mathrm{SSE}_k}{\mathrm{SSE}_{k-1}/(n-k-1)} \sim F_{1,n-k-1},$$

the F-distribution with 1 and $n-k-1$ degrees of freedom. We compute the F statistic at each degree and stop the IRF procedure if the corresponding p-value

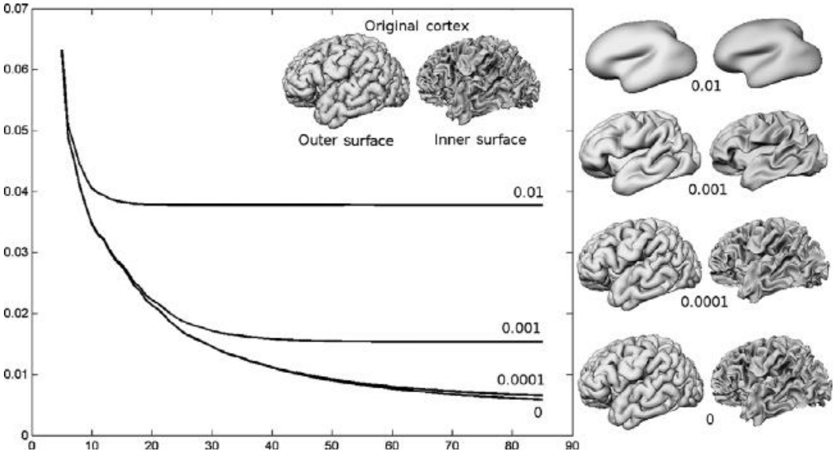


Figure 12.3 Plots of the root mean square deviation (RMSD) for the weighted spherical harmonic (SPHARM) representation of brain surface coordinates with varying σ (0.01, 0.001, 0.0001, 0). When $\sigma = 0$, we have the traditional spherical harmonic representation. For $\sigma > 0$, the representation is equivalent to isotropic heat diffusion with diffusion time σ. The cortical surfaces correspond to the 85th degree expansion. As $\sigma \to 0$, the weighed representation converges to the traditional SPHARM (Chung et al., 2008a).

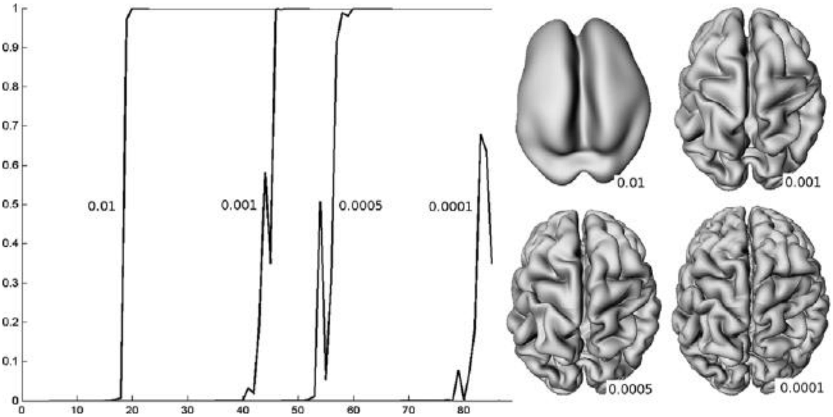


Figure 12.4 Cortical thickness projected onto the average outer cortex for various t and corresponding optimal degree: $k = 18 (t = 0.01), k = 42 (t = 0.001), k = 52 (t = 0.0005), k = 78 (t = 0.0001)$. The average cortex is constructed by averaging the coefficients of the weighted SPHARM. The highly noisy first image shows thickness measurements obtained by computing the distance between two triangle meshes.

first becomes bigger than the prespecified significance α, which is usually set at 0.05 (Figures 12.3 and 12.4).

12.3 Spectral Decomposition with Different Bases

Consider two groups of subjects. Suppose group 1 gives the connectivity matrix C^1 and group 2 gives the connectivity matrix C^2. The problem of applying the spectral decomposition to each connectivity matrix separately is that they produce different bases. Instead of doing spectral decomposition separately for each group, we combine the subjects. Let C be the connectivity matrix obtained from the combined group. The spectral decomposition gives

$$C = \sum_{j=1}^{p} \lambda_j v_j v_j^\top. \tag{12.11}$$

Obviously we cannot expand connectivities C^1 and C^2 with basis v_j but we can approximate them in the subspace spanned by $\{v_j\}$. We can estimate coefficients α_j and β_j of the expansions

$$C^1 = \sum_{j=1}^{p} \alpha_j v_j v_j^\top$$

$$C^2 = \sum_{j=1}^{p} \beta_j v_j v_j^\top.$$

The coefficients can be estimated sequentially one at a time using the IRF algorithm (Chung et al., 2008a). First we estimate α_1 by solving

$$C^1 = \alpha_1 v_1 v_1^\top. \tag{12.12}$$

The solution of the matrix equation (12.12) is

$$\widehat{\alpha}_1 = v_1^\top C^1 v_1.$$

Subsequently, we solve for α_2:

$$C^1 - \widehat{\alpha}_1 v_1 v_1^\top = \alpha_2 v_2 v_2^\top.$$

α_2 is similarly estimated as

$$\widehat{\alpha}_2 = v_2^\top [C^1 - \widehat{\alpha}_1 v_1 v_1^\top] v_2.$$

The process is iteratively continuous until all the coefficients are estimated. Then C^1 is approximately decomposed using basis $\{v_j\}$. However, this approach may *not* work if the the norm on the residual matrix does not decrease. It is necessary to make the residual matrix decreases.

12.4 Spectral Permutation

Suppose we have two groups of subjects. One consists of normal controls and the other is most likely clinical population. Suppose we have connectivity matrices C^1 and C^2. Consider their spectral decompositions:

$$C^1 = \sum_{j=1}^{p} \lambda_j^1 v_j^1 {v_j^1}^\top$$

$$C^2 = \sum_{j=1}^{p} \lambda_j^2 v_j^2 {v_j^2}^\top.$$

Then we will permute the set of eigenvalues between two matrices. An example of permutation is

$$C_\sigma^1 = \lambda_1^2 v_1^1 {v_1^1}^\top + \sum_{j=2}^{p} \lambda_j^1 v_j^1 {v_j^1}^\top$$

$$C_\sigma^2 = \lambda_1^1 v_1^2 {v_1^2}^\top + \sum_{2=1}^{p} \lambda_j^2 v_j^2 {v_j^2}^\top$$

where only the first eigenvalues λ_1^1 and λ_1^2 are permuted. σ is an index to indicate such permutation. Under the null assumption of the equivalence of connectivity matrices, we have

$$H_0 : \lambda_j^1 = \lambda_j^2 \text{ for all } j.$$

Therefore, we may permute the whole eigenvalue and eigenvector together.

There will be total $\binom{2p}{p}$ permutations. However, we may be most likely to permute the first q largest eigenvalues $\lambda_1^1, \cdots, \lambda_q^1$ and $\lambda_1^2, \cdots, \lambda_q^2$ and leave out the high-frequency components. q should be determined empirically. Given n images in each group, there are total $\binom{2n}{n}$ permutations. If we choose q to be far smaller than n, we also have additional computational speed gain.

12.5 Karhunen–Loève Expansion

Karhunen–Loève expansion can be used to represent correlation and covariance matrices continuously using continuous basis functions. Let $\mathcal{G}$ be the space of zero mean Gaussian random fields in some manifold $\mathcal{M} \subset \mathbb{R}^d$ with inner product

$$\langle X, Y \rangle = \mathbb{E} \int_{\mathcal{M}} X(x)Y(x)\, dx$$

with $\|X\| < \infty$. This is basically the integral of the cross-covariance function between fields X and Y. $\mathcal{G}$ can be shown to be a separable Hilbert space by finding countable orthonormal basis in $\mathcal{G}$ (Adler and Taylor, 2007).

Theorem 12.1 *(Karhunen–Loéve expansion) For a mean zero Gaussian random field $Z(x)$ in $\mathcal{G}$ with mean square continuity over a bounded domain $\mathcal{M} \subset \mathbb{R}^d$, there exist independent mean zero Gaussian random variables $Z_i \sim N(0, \sigma_i^2)$ and orthonormal bases ψ_i such that*

$$Z(x) = \sum_{i=0}^{\infty} Z_i \psi_i(x). \tag{12.13}$$

If $\mathbb{E}Z(x) \neq 0$ for some x, we can always center the field by translating toward the sample mean. The basis ψ_i is orthonormal in $\mathcal{M}$ such that $\langle \psi_i, \psi_j \rangle = \delta_{ij}$. Let $\sigma_i^2 = \mathbb{E}Z_i^2 < \infty$. Since $\mathbb{E}Z(x) = 0$, the covariance function of $Z(x)$ is given by

$$\begin{aligned} R(x, y) &= \mathbb{E} \sum_{i,j=0}^{\infty} Z_i \psi_i(x) Z_j \psi_i(y) \\ &= \sum_{i=0}^{\infty} \sigma_i^2 \psi_i(x) \psi_i(y). \end{aligned} \tag{12.14}$$

The covariance function of a zero mean Gaussian field completely characterizes the field itself. Obviously R has to be symmetric to be expressible in this fashion. This fact is related to Mercer's theorem (Conway, 1990).

If $f_j(x)$ is the realization of the random field Z, the parameters σ_i^2 can be estimated by matching the moment in the following fashion. From (12.14), we have

$$R(x, x) = \sum_{i=0}^{\infty} \sigma_i^2 \psi_i^2(x).$$

If we assume f_j are also centered, the left-hand side is the variance field, which can be estimated using the sample variance field:

$$\frac{1}{n}\sum_{j=1}^{n} f_j^2(x).$$

Then we first estimate the parameter σ_0 by solving

$$\frac{1}{n}\sum_{j=1}^{n}\int_{\mathcal{M}} f_j^2(x)\,dx = \sigma_0^2\int_{\mathcal{M}}\psi_0^2(x)\,dx = \sigma_0^2.$$

Once we estimated σ_0 as $\widehat{\sigma_0}$, the next parameter σ_1 is then estimated by solving

$$\frac{1}{n}\sum_{j=1}^{n}\int_{\mathcal{M}} f_j^2(x) - \widehat{\sigma_0}^2\int_{\mathcal{M}}\psi_0^2(x)\,dx = \sigma_1^2\int_{\mathcal{M}}\psi_1^2(x)\,dx.$$

The process is iteratively performed until we obtain a sufficiently high-degree representation. The estimation process is similar to the iterative residual fitting algorithm. The Karhunen–Loève expansion is just one example of many possible orthonormal expansions of a function.

12.5.1 Mercer's Theorem

For continuous symmetric kernel R, which can be taken as a continuous connectivity matrix, define linear operator $\mathcal{L} : L^2(\mathcal{M}) \to L^2(\mathcal{M})$ as

$$\mathcal{L}f(x) = \int_{\mathcal{M}} R(x,y)f(y)\,dy.$$

This is a compact self-adjoint operator. The linear operator yields unique countable eigenvalues σ_i^2 and orthonormal eigenfunctions ψ_i of the operator $\mathcal{L}$ such that

$$\mathcal{L}\psi_i = \sigma_i^2\psi_i \tag{12.15}$$

with $\sigma_\infty^2 = 0$. Equation (12.15) is a Fredholm equation of the first kind, and ψ_i and σ_i^2 can be estimated numerically if the kernel $R(x,y)$ is given (Arfken, 2000). We will order the eigenvalues such that

$$\sigma_0^2 > \sigma_1^2 > \sigma_2^2 > \cdots .$$

Then any function $f \in L^2(\mathcal{M})$ can be represented as

$$f = \sum_{i=1}^{\infty}\langle\psi_i, f\rangle\psi_i.$$

Subsequently, the operator $\mathcal{L}$ has a spectral representation

$$\mathcal{L}f = \sum_{i=0}^{\infty} \sigma_i^2 \langle \psi_i, f \rangle \psi_i.$$

Then Mercer's theorem states that kernel R is expressed as

$$R(x,y) = \sum_{i=0}^{\infty} \sigma_i^2 \psi_i(x)\psi_i(y). \tag{12.16}$$

A special case of Mercer's theorem is when R is the heat kernel given by

$$R(x,y) = \sum_{i=0}^{\sigma} e^{-\lambda_i t} \psi_i(x)\psi_i(y).$$

The corresponding linear operator $\mathcal{L}$ is the *heat kernel smoothing* operator defined as

$$\mathcal{L}f(x) = \sum_{i=0}^{\sigma} e^{-\lambda_i t} \langle \psi_i, f \rangle \psi_i(x).$$

Mercer's theorem can be proved using the following argument given in Courant and Hilbert (1953). Suppose we fix y. Then from the Weierstrass's approximation theorem, for continuous function, $R(x,\cdot) = \sum_{i=0}^{\infty} \alpha_i(x)$ for some basis functions α_i uniformly. Now fix x and we have

$$R(x,y) = \sum_{i=0}^{\infty} \alpha_i(x) \sum_{j=0}^{\infty} \beta_j(y) \tag{12.17}$$

$$= \sum_{i,j=0}^{\infty} \alpha_i(x)\beta_j(y). \tag{12.18}$$

Again, β_j are basis functions. For basis $\alpha_1, \cdots, \alpha_p$ and $\beta_1, \cdots, \beta_p$, they can be rewritten as a linear combination of our orthonormal basis ψ using the Gram–Schmidt orthogonalization. So some algebraic manipulation can show that

$$R(x,y) = \sum_{i,j=0}^{\infty} c_{ij} \psi_i(x)\psi_j(x) \tag{12.19}$$

for some c_{ij}. Another way of looking at this problem is by noting that $\psi_i(x)\psi_j(y)$ forms an orthonormal basis for $\mathcal{M} \otimes \mathcal{M}$. Then for the covariance function $R(x,y) \in L^2(\mathcal{M} \otimes \mathcal{M})$, we immediately have the series expansion (12.19).

We will identify c_{ij}, using the definition of $\mathcal{L}$. We have

$$\begin{aligned}\mathcal{L}\psi_k(x) &= \sum_{i,j=0}^{\infty} c_{ij}\psi_i(x)\int_{\mathcal{M}} \psi_k(y)\psi_j(y)\,dy\\ &= \sum_{i,j=0}^{\infty} c_{ij}\psi_i(x)\delta_{kj}\\ &= \sum_{i=0}^{\infty} c_{ik}\psi_i(x). \qquad (12.20)\end{aligned}$$

We need to equate (12.20) to $\sigma_k^2\psi_k(x)$. The only way it is satisfied for all x and k is when $c_{kk} = \sigma_k^2$ and $c_{ik} = 0$ for $i \neq k$. Hence we proved the statement of Mercer's theorem (12.16).

12.6 Vandermonde Matrix Expansion

We are interested in representing connectivity matrices C^1 and C^2 in terms of an expansion involving another connectivity matrix C exactly. This requests three steps. Suppose $D_1 = (\lambda_i^1), D_2 = (\lambda_i^2), D = (\lambda_i) \in \mathbb{R}^{p\times p}$ are diagonal matrices consisting of eigenvalues

$$\lambda_i \neq \lambda_j, \text{ if } i \neq j. \qquad (12.21)$$

In practice, this is not necessarily a strong assumption. If the connectivity matrix takes continuous values, the probability of such event is zero. If the connectivity matrix takes discrete values, it may be possible to have some of diagonal entries be identical. Then we may simply add negligibly small numbers and make them unique.

Consider the Vandermonde matrix

$$A = \begin{pmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{p-1}\\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{p-1}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 1 & \lambda_p & \lambda_p^2 & \cdots & \lambda_p^{p-1}\end{pmatrix}. \qquad (12.22)$$

It can be shown that the determinant of A is

$$\det A = \prod_{1\le i\le j\le p} (\lambda_j - \lambda_i).$$

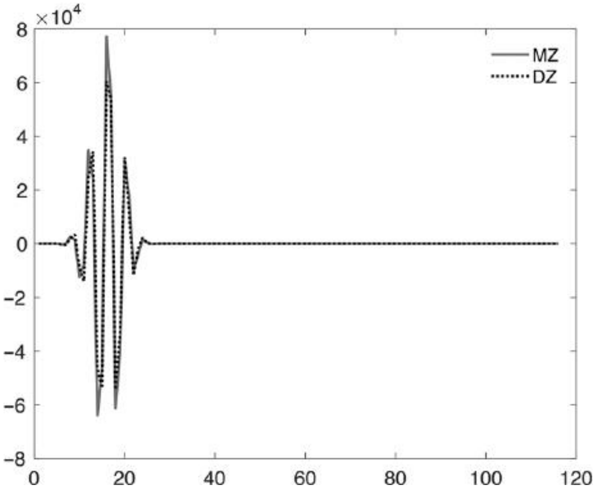


Figure 12.5 The 116 coefficients obtained in the Vandermonde matrix expansion for MZ- (solid line) and DZ-twins (dotted line). Almost all coefficients are negligible except the first 36 coefficients. The remaining 80 coefficients are smaller than 0.001 in absolute magnitude. With 36 coefficients, the reconstruction error is smaller than 0.0029 elementwise in L_2 norm in average.

Thus, from condition (12.21), A is invertible. Let $\Lambda_1 = (\lambda_1^1, \lambda_2^1, \cdots, \lambda_p^1)^\top$ be the vector of diagonal entries of D. Let $\mathbf{x} = (x_1, x_2, \cdots, x_p)^\top$ be the solution of

$$\Lambda_1 = A\mathbf{x}.$$

Then we have

$$\lambda_i^1 = x_1 + \lambda_i x_2 + \cdots + \lambda_i^{p-1} x_p.$$

Identifying the expansion coefficients $\mathbf{x}$ requires inverting the Vandermonde matrix (Figure 12.5). The exact analytical form of the inverse of the Vandermonde matrix is known. After identifying $\mathbf{x}$, we have

$$D_1 = \sum_{k=1}^{p} x_k D^{k-1}.$$

Since C^1 and C^2 are symmetric, there exist orthogonal matrices Q_1 and Q_2 such that

$$Q_1^\top C^1 Q_1 = D_1, \quad Q_2^\top C^2 Q_2 = D_2, \quad Q^\top C Q = D. \tag{12.23}$$

Note that $Q^\top Q = QQ^\top = I$ and $Q^\top C^{k-1} Q = D^{k-1}$. Hence, we have

$$\begin{aligned} C^1 = Q_1 D_1 Q_1^\top &= Q_1 \Big[\sum_{k=1}^{p} x_k D^{k-1} \Big] Q_1^\top \\ &= Q_1 \left[\sum_{k=1}^{p} x_k Q^\top C^{k-1} Q \right] Q_1^\top \\ &= \sum_{k=1}^{p} x_k P_1^\top C^{k-1} P_1, \end{aligned}$$

where $P_1 = Q(Q_1)^\top$. Note $P_1 P_1^\top = P_1^\top P_1 = I$. Thus, we have expansion

$$P_1 C^1 P_1^\top = \sum_{k=1}^{p} x_k C^{k-1}.$$

Similarly we have

$$P_2 C^2 P_2^\top = \sum_{k=1}^{p} y_k C^{k-1}$$

with $P_2 = Q(Q_2)^\top$. The expansion is exact within the numerical accuracy of singular value decomposition.

Example 12.1 *If the assumption* (12.21) *is not satisfied, we cannot represent C^1 using C. Here is a counterexample. Consider corresponding diagonal matrices of eigenvalues:*

$$D_1 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{12.24}$$

Since A is not invertible, equation $A\mathbf{x} = \Lambda_1$ has no solution. Thus, D^1 can not be represented in terms of the expansion of C. Also, if two eigenvalues are too close to each other, the condition number will be high and we have an ill-conditioned problem. The expansion may not be accurate.

Example 12.2 *Consider following connectivity matrices*

$$C_1 = \begin{pmatrix} 4 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 3 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}. \tag{12.25}$$

The corresponding diagonal and orthonormal matrices are

$$D_1 = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 2.38 & 0 \\ 1 & 0 & 4.62 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}.$$

$$Q_1 = \begin{pmatrix} 0 & 0.53 & 0.85 \\ -1 & 0 & 0 \\ 0 & -0.85 & 0.53 \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

The Vandermonde matrix is

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix}.$$

Solving $A\mathbf{x} = \Lambda_1$, *we have* $\mathbf{x} = (3.47, -2.40, 0.93)^\top$. *Subsequently, we can represent* C_1 *as the series expansion exactly.*

Example 12.3 *The Vandermonde matrix expansion is exact if the inverse of Vandermonde matrix is well defined. However, in practice it may be difficult to make the matrix well conditioned. Consider the Spearman correlation matrices* C^1 *and* C^2 *from MZ- (solid red) and DZ-twins (dotted black) obtained from a DTI study (Chung et al., 2018b) (Figure 12.2). Let* C *be the Spearman correlation matrix obtained from by combining both MZ- and DZ-twins. For better estimation of the coefficients* $\mathbf{x}$, *we expanded the differences* $C^1 - C$ *and* $C^2 - C$ *with respect to* C. *Then added* C *after the expansion. The results are given in Figure 12.6.*

12.6.1 Transition Matrix

If we ignore the negative entries, the connectivity matrices can be interpreted as *transition matrices* for Markov chains with an additional normalization. Given connectivity matrix $C = (c_{ij})$, C is considered as a proper transition matrix if all the entries are nonnegative, i.e., $c_{ij} \geq 0$ and doubly stochastic such that

$$\sum_{i=1}^{p} c_{ij} = \sum_{j=1}^{p} c_{ij} = 1.$$

In case C is not doubly stochastic, we can make it doubly stochastic as follows:

$$C \to ACB,$$

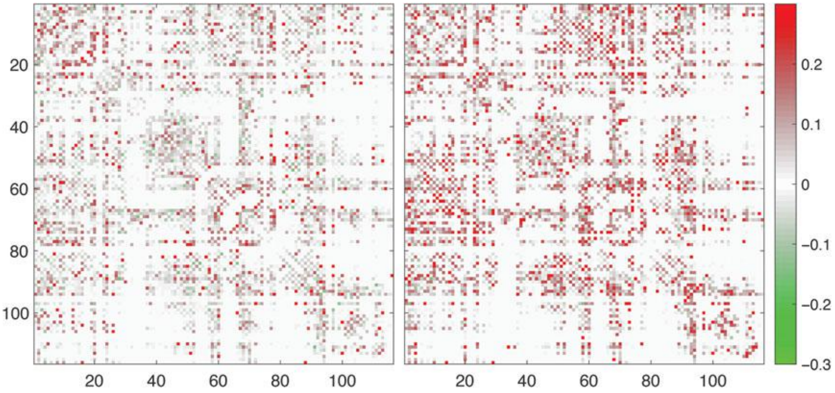


Figure 12.6 Reconstruction of DZ-twin (left) and DZ-twin correlation matrices using 36 coefficients in the Vandermonde matrix expansion. Compared to the original correlation matrices in Figure 12.5, we obtained almost similar patterns. The reconstruction error is smaller than 0.0029 elementwise in L_2 norm in average.

where $A = (a_{ij})$ and $B = (b_{ij})$ with

$$a_{ij} = 1 \Big/ \sum_{k=1}^{p} c_{ik}$$

$$b_{ij} = 1 \Big/ \sum_{k=1}^{p} c_{kj}$$

Assuming such constraints are satisfied, the Vandermonde matrix expansion may be viewed as a regression problem on graphs such that we are regressing other connectivity matrix C^1 as an unknown linear combination of transition matrices, i.e.,

$$g(C^1) = x_1 I + x_2 C + \cdots + x_p C^{p-1},$$

where $g : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ is some nonlinear matrix function that connects C^1 to the powers of C. In our application, $g(C^1) = P_1 C^1 P_1^\top$.

Since entries of C are between 0 and 1, the entries of C^i get smaller as i increases. Thus, higher power of C can be treated analogous as high-frequency bases in Fourier series expansions.

12.7 The Space of Positive Definite Symmetric Matrices

Recently in relation to brain network analysis and in diffusion tensor imaging, positive definite symmetric (PDS) matrices have become fundamental objects of interest. Correlation and covariance matrices at the population level should be all considered as PDS even though we often obtain nonnegative definite matrices. The usual Euclidean geometry does not apply in the space of PDS. Let $\mathcal{P}_m \subset \mathbb{R}^{m(m+1)/2}$ be the space of positive definite symmetric matrices of size $m \times m$. Due to the geometric nature of PDS, differential geometric approaches are useful in quantifying the space (Kreyszig, 1959; Boothby, 1986; do Carmo, 1992; Joshi et al., 1995).

Suppose we have manifold $\mathcal{P}_m$ which is a smooth twice-differentiable manifold embedded in $\mathbb{R}^{m(m+1)/2}$. We have a parameterization

$$X(u) : \mathcal{N} \to \mathcal{P}_m,$$

where all partial derivatives of X up to the second order are continuous in some domain $\mathcal{N}$, and partial derivatives $X_j = \partial X/\partial u_j$ form a basis for the tangent space $T_Y(\mathcal{P}_m)$ at point $Y \in \mathcal{P}_m$. For this to happen, we need $X_i(u) \times X_j(u) \neq 0$ for any $u \in \mathcal{N}$ and $i \neq j$. The coefficients $g_{ij} = \langle X_i, X_j \rangle$ are called the *Riemannian metric tensor* and they measure the amount of deviation from the Cartesian coordinates.

12.7.1 Laplace–Beltrami Operator

The gradient ∇_X of a function F on the tangent plane $T_Y(\mathcal{P}_m)$ is defined as

$$\nabla_X F = \sum_{i,j} g^{ij} \frac{\partial F}{\partial u^j} X_i, \tag{12.26}$$

where $(g^{ij}) = g^{-1}$. The divergence $\nabla_X \cdot$ of a vector field $V = \sum_i V^i X_i$ is then

$$\nabla_X \cdot V = \frac{1}{|g|^{1/2}} \sum_i \frac{\partial}{\partial u^i} (|g|^{1/2} V^i),$$

where $|g|$ is the determinant of g. The generalized Laplacian called the *Laplace–Beltrami operator* Δ_X corresponding to parameterization X is then defined as the divergence of the gradient operator (Kreyszig, 1959; Marsden and Hughes, 1983) such that

$$\Delta_X F = \nabla_X \cdot (\nabla_X F) = \frac{1}{|g|^{1/2}} \sum_{i,j} \frac{\partial}{\partial u^i} \Big(|g|^{1/2} g^{ij} \frac{\partial F}{\partial u^j} \Big). \tag{12.27}$$

For the derivation of the Laplace–Beltrami operator without using differential geometry, one may approach the problem in terms of a curvilinear coordinate transform (Courant and Hilbert, 1953). The most important intrinsic property of the Laplace–Beltrami operator is that it is independent of the parameterization of $\mathcal{P}_m$. If $\widetilde{X} = X \circ \Phi$ is another parameterization, we have

$$\Delta_{\widetilde{X}} \widetilde{F} = \widetilde{\Delta_X F}.$$

However, one should be careful in choosing a proper parameterization, which stabilizes the numerical computation and minimizes the variances of errors in estimating the Laplace–Beltrami operator.

The Laplace–Beltrami operator is *self-adjoint* so if F and G are twice differentiable functions in $\mathcal{P}_m$,

$$\begin{aligned}\int_{\mathcal{P}_m} G \Delta F \, d\mu &= \int_{\mathcal{P}_m} F \Delta G \, d\mu \\ &= -\int_{\mathcal{P}_m} \langle \nabla F, \nabla G \rangle \, d\mu,\end{aligned}$$

where the inner product is defined as

$$\langle \nabla F, \nabla G \rangle = \sum_{ij} g^{ij} \frac{\partial}{\partial u^i} F \frac{\partial}{\partial v^i} G$$

and the area element $d\mu = \sqrt{\det g} \, du$ (Grigoryan, 1999).

The *conformal coordinates* are defined as a coordinate system whose metric is given by

$$ds^2 = \sum_i \lambda (du^i)^2$$

for some function $\lambda = \lambda(u)$. With respect to the conformal coordinates, the Laplace–Beltrami operator is simplified to

$$\Delta_X = \frac{1}{\lambda} \sum_i \frac{\partial}{\partial u^i}.$$

For an arbitrary smooth surface and a fixed point p, we can always find conformal coordinates such that $Y = X(u)$ and $\lambda(u) = 1$ (do Carmo, 1992). Therefore, if we find a conformal coordinate system at each $Y \in \mathcal{P}_m$, the computation of the Laplace–Beltrami operator at $Y = X(u)$ can be simplified to the usual Euclidean Laplacian at u.

12.7.2 Eigenfunctions in PDS

The eigenfunctions and the eigenvalues of the Laplace–Beltrami operator have been used in many different contexts in literature (Lévy, 2006; Zhang et al., 2010). Qiu et al. (2006) constructed splines on a brain cortical surface with a boundary using the eigenfunctions. Seo et al. (2010) constructed the heat kernel as a series expansion of eigenfunctions and formulated diffusion as heat kernel smoothing. Vallet and Lévy (2008) used the eigenfunctions to analytically formulate geometric filtering problems. Dong et al. (2006) proposed a quadrangular remeshing of surfaces using the evenly distributed extrema of eigenfunctions. The eigenfunctions are used in shape analysis (Reuter et al., 2009) and shape segmentation and registration (Reuter, 2010).

Since the Laplace–Beltrami operator is self-adjoint and elliptic, we have discrete eigenvalues

$$0 = \lambda_0 \le \lambda_1 \le \lambda_2 \le \cdots$$

and the corresponding eigenfunctions ψ_j satisfying

$$\Delta \psi_j = \lambda_j \psi_j. \tag{12.28}$$

The eigenfunctions ψ_j form the orthonormal basis in $L^2(\mathcal{P}_m)$, the space of square integrable functions in $\mathcal{M}$. Numerically, eigenfunctions can be made into orthonormal by scaling the eigenvalues. Eigenfunctions are often not known analytically unless the underlying manifold is algebraically given.

For $Y = (y_{ij}) \in \mathcal{P}_m$, let $dY = (dy_{ij})$. Following Maass (1955), we put the following metric on $\mathcal{P}_m$:

$$(ds)^2 = \operatorname{tr}\Big((Y^{-1}dY)^2\Big).$$

Vectorize $n = m(m+1)/2$ unique entries of Y as $(x_1, x_2, \cdots, x_n)'$ and write ds^2 in the standard quadratic form as

$$(ds)^2 = \sum g_{ij} dx_i dx_j.$$

For $\mathcal{P}_m$, this can be more compactly written as follows. Define the matrix of differential operators ∂ as

$$\partial_Y = \Big(\frac{1}{2}(1 + \delta_{ij})\frac{\partial}{\partial y_{ij}}\Big),$$

where δ_{ij} is Kronecker's delta. With this operator, the Laplace–Beltrami (LB)-operator Δ in the local coordinates y_{ij} is given by (Richards, 1985; Haff et al., 2011)

$$\Delta = \operatorname{tr}(Y\partial_Y)^2. \tag{12.29}$$

Note that the Laplacian in the coordinates of the eigenvalues of y has more complicated from James (1968).

Consider eigensystem

$$\Delta\psi_k(Y) = -\lambda_k\psi_k(Y).$$

The eigenfunction of the Laplacian (12.29) is difficult to compute in practice and involves zonal spherical functions (Richards, 1985, 2011). There are currently no available tools for computing these functions. However, we can numerically solve the system using the finite difference scheme by discretizing the Laplacian (Chung and Taylor, 2004; Chung et al., 2015b). Since the eigenmatrix ψ_j is othonormal, any positive definite symmetric matrices $C \in \mathcal{P}_m$ can be expressed as

$$C = \sum_j c_j\psi_j$$

where $c_j = \int_{\mathcal{P}_m} C(Y)\psi_j(Y)\, d\mu$.

Example 12.4 *Consider $\mathcal{P}_1 = \mathbb{R}^+$, the positive real line. Note that the Laplacian is paramaterization invariant. Let $y = e^x$ be the parameterization of $\mathcal{P}_1$. It maps $\mathbb{R}$ to $\mathbb{R}+$. Then $dy = ydx$ and with respect to the orginal coordinates y, we obtain*

$$\Delta = \left(\frac{d}{dx}\right)^2 = \left(y\frac{d}{dy}\right)^2 = y\frac{d}{dy} + y^2\frac{d^2}{dy^2}. \tag{12.30}$$

Laplacian (12.30) *in $\mathcal{P}_1$ differs from the usual Laplacian $\frac{d^2}{dy^2}$ for the whole real line. This additional algebraic complexity of the Laplacian makes the computation of eigenfunctions of even 1D cases* (12.30) *complicated. In fact, we need to solve*

$$y\frac{d}{dy}\psi_j(y) + y^2\frac{d^2}{dy^2}\psi_j(y) = \lambda_j\psi_j(y). \tag{12.31}$$

In practice, it is easier to simply discretize the differential equation (12.31) and solve it using the finite difference method.

13
Dynamic Network Models

Spontaneous fluctuations of neural signals are hallmark of resting-state functional magnetic resonance imaging (rs-fMRI) (Allen et al., 2014). Recently, brain network studies have shifted their focus toward dynamic characterization of coactiviation patterns. This results in the recognization of intrinsic connectivity patterns including the default-mode network in rs-fMRI (Greicius et al., 2004). Compared to task-based fMRI, in rs-fMRI dynamic patterns are more prominent (Allen et al., 2014). Thus, it is necessary to model rs-fMRI using various dynamic network models.

A stable connectivity pattern that behaves like the backbone of networks through fluctuating functional connectivity needs to be identified. The change time points where the simultaneous fluctuations occur and disappear should be identified as well. Such a complex dynamic pattern is difficult to model completely with a single complicated statistical model. Many existing graph theory–based static brain network models largely do not take into account the presence of spontaneous temporal fluctuations. The dynamic brain network analysis has its roots in modeling effective connectivity in fMRI, where they looked at the causal influence of one neuronal system exerts over another (Friston et al., 1993b; Friston, 1994; Marrelec et al., 2009). Traditionally the effective connectivity has been modeled using structural equation modeling (SEM) (Mclntosh and Gonzalez-Lima, 1994; Bullmore and Sporns, 2009) and dynamic causal modeling (DCM) (Friston et al., 2003a,b). Although DCM and SEM are implemented in brain image analysis tools such as SPM and AFNI, the methods were often applied to a small number of nodes due to computational complexity.

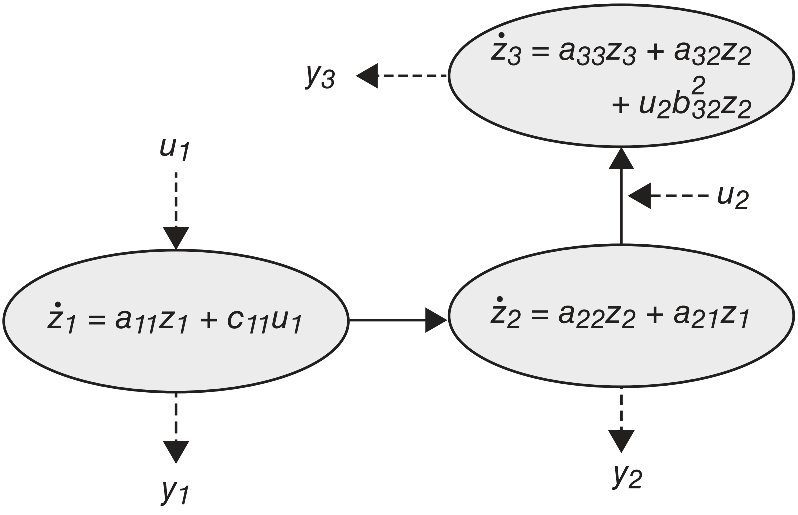
13.1 Dynamic Causal Model

The DCM (Friston et al., 2003; Penny et al., 2004; David et al., 2006) has been increasingly popular in fMRI connectivity studies. DCM is an effort to model neuronal response as a physical system using a collection of first-order differential equations. DCM models the dynamic change of neuronal response in fMRI.

Let $z = (z_1, \cdots, z_n)'$ be the neuronal response at n nodes and $u_j = (u_{j1}, \cdots, u_{jm})'$ be the jth input signal. Then the neuronal activity is modeled as

$$\frac{dz}{dt} = Az + \sum_{j=1}^{m} u_j B^j z + Cu.$$

Friston et al. (2003) termed $A = (a_{ik})$ and $B^j = (b^j_{ik})$ as the latent and induced connectivity matrices. A and B^j matrices measure the dependency between nodes. Figure 13.1 shows the schematic of DCM with three nodes. The temporal change in the neuronal response is basically modeled using the measurements obtained in all other nodes simultaneously. DCM is implemented in the SPM extension package.



$\dot{z}_3 = a_{33}z_3 + a_{32}z_2$
$+ u_2 b^2_{32} z_2$
y_3
u_1
u_2
$\dot{z}_1 = a_{11}z_1 + c_{11}u_1$
$\dot{z}_2 = a_{22}z_2 + a_{21}z_1$
y_1
y_2

Figure 13.1 The schematic of DCM on three nodes with two inputs. The output y_j is somehow obtained from the neuronal response z_j.

13.1.1 Dynamic Sparse Network Model

The main limitation of DCM or any type of linear model is that they are usually inappropriate for applications to whole brain regions mainly due to the *small-n large-p problem* (Friston et al., 1995; Schäfer and Strimmer, 2005; Valdés-Sosa et al., 2005; Lee et al., 2011c; Chung et al., 2013). Specifically, the number of nodes p a is substantially larger than the number of images n, which results in an improper estimation of the covariance between nodes. The $small - n\ large - p$ problem can be remedied by using sparse network approaches, which regularize the estimated rank deficient covariance matrix with additional sparse penalties (Avants et al., 2010; Huang et al., 2010; Lee et al., 2011c; Mazumder and Hastie, 2012; Chung et al., 2013). So far sparse network models are mainly applied to modeling static networks obtained from the resting state fMRI, DTI (Chung et al., 2015a), or PET (Lee et al., 2011c).

The sparse network framework introduced to handle static networks can be adapted to handle temporally changing dynamic networks by introducing time dependency in the model. We present a *dynamic sparse network model* for modeling temporally changing, possibly task-dependent, networks. The system of linear equations characterizes the network for a population at a specific fixed time point. For statistical data, we set up a linear model on measurement z_j at node j:

$$z_j = \sum_{k \neq j} \beta_{jk} z_k + \epsilon_k,$$

where β_{jk} measures the connectivity strength between nodes j and k (Chung, 2012). Then we are interested in testing the significance of

$$H_0 : \beta_{jk} = 0 \text{ vs. } H_1 : \beta_{jk} = 0.$$

For a temporally changing network, the measurements z_j and parameter matrix $\beta = (\beta_{jk})$ are also considered as temporally changing. Thus the change should be characterized by the change of β over time. So we start with the following dynamic linear model:

$$z_j(t) = \sum_{k \neq j} \beta_{jk}(t) z_k(t) + \epsilon_k(t), \tag{13.1}$$

where $z_j(t)$ is a sufficiently smooth neuronal response and $\epsilon_k(t)$ is a smooth Gaussian random field. Since (13.1) is a linear model, the parameter can be estimated using the least squares method. Then we can sparsely estimate β_{jk} by adding an additional sparse term in the least squares. This results in the LASSO-type sparse parameter estimation $\widehat{\beta}_{jk}$.

By differentiating (13.1) with respect to t, we have a dynamic model at node j:

$$\frac{dz_j}{dt} = \sum_{k \neq j} \beta_{jk} \frac{dz_k}{dt} + \sum_{k \neq j} \frac{d\beta_{jk}}{dt} z_k + \frac{d\epsilon_j}{dt} \tag{13.2}$$

If there is no temporal change in the network, we expect $\frac{d\beta_{jk}}{dt}$ to vanish. So the hypothesis of interest is

$$H_0 : \frac{d\beta_{jk}}{dt} = 0 \text{ vs. } H_1 : \frac{\beta_{jk}}{dt} \neq 0.$$

Therefore, it is necessary to estimate the derivative $\frac{d\beta_{jk}}{dt}$ somehow. We can estimate the derivatives using a two-step estimation technique.

Since we already have the sparse estimate $\widehat{\beta}_{jk}$ for the parameters, (13.2) can be written as

$$\frac{dz_j}{dt} - \sum_{k \neq j} \widehat{\beta}_{jk} \frac{dz_k}{dt} = \sum_{k \neq j} \frac{d\beta_{jk}}{dt} z_k + \frac{d\epsilon_j}{dt}. \tag{13.3}$$

The left-hand side is known and $\frac{d\beta_{jk}}{dt}$ is further estimated similarly. Since we already have LASSO estimate $\widehat{\beta}_{jk}$ for the parameters, the only unknown quantity $\frac{d\beta_{jk}}{dt}$ can be further estimated again using LASSO-type sparse methods.

13.2 Dynamic Time Series Models

Correlation and covariance matrices can be often treated as manifold-valued data, where the data are defined in the space of positive definite symmetric (PDS) matrices with a Riemannan metric. In this section, we will explain dynamic models of manifold-valued data with a focus on dynamic PDS structures from nonstationary multivariate time series. The models can capture how connectivity dynamically changes over time and thus can be used to evaluate evolutionary dynamics of functional brain networks.

There is extensive literature on nonstationary spectra and covariances (Ombao et al., 2005; Ombao and Van Bellegem, 2008; Ombao et al., 2017). However, the existing work only indirectly defines the connectivity structures from the time series representations. Better models would be to directly model the time series of manifold-valued data that respect the underlying geometric structure of PDS.

13.2.1 Time Series Models on Tangent Space

Some preliminary ideas of building the autoregressive (AR) model to manifolds were described in Xavier and Manton (2006), but this work only provides a sketch of an estimation procedure in the simplified setting of AR processes on the unit circle. We introduce a nontrivial generalization of the AR model specifically for PDS matrices, along with explicit estimation procedures and statistical methods suitable for practical application. The methods can be used to examine the evolution of brain connectivity over time.

Consider a time series $Y_1, Y_2, \cdots, Y_T$. We can model the increment between Y_{t-1} and Y_t, defined as

$$\Delta_t = \log_{Y_{t-1}} Y_t = \log Y_t - \log Y_{t-1},$$

with an autoregressive model in the tangent space of Y_{t-1}, and map back to the manifold after incrementing using the exponential map, i.e.,

$$Y_t = \exp_{Y_{t-1}} \Delta_t.$$

The proposed order r AR model for Y_t is specified by

$$\Delta_t = \sum_{i=1}^{r} A_i \mathcal{P}_{Y_{t-i-1}, Y_{t-1}} (\Delta_{t-i}) + W_t, \qquad Y_t = \mathrm{Exp}_{Y_{t-1}}(\Delta_t),$$

where W_t is zero mean noise, and $\mathcal{P}_{a,b}$ is the parallel transport operator, which is needed since the coefficient matrices $A_1, \cdots, A_p$ are in different tangent spaces.

Although this model is based in the classical AR model, it is a nontrivial extension since standard computational methods are not directly applicable for general manifolds. Numerically, since we are dealing with symmetric matrices, we can map Δ_t to its vectorized upper triangle, including the diagonals, and solve the system of time series as if they are in the Euclidean space.

13.2.2 Time Series Models on Manifolds

As an alternative to the tangent space model in the previous section, we can set up an exact time series model on a manifold. While it can be attractive to construct the time series model on the tangent space due to computational ease, such a model necessarily results in approximation error. Intrinsic time series models on the manifold will have improved accuracy over the tangent space methods due to using the orthonormal basis of the manifolds. With this approach, it is no longer necessary to project to the tangent space to do the computation and then project back into the manifold. Let $Y_1, \cdots, Y_T \in \mathbb{P}$, the space of symmetric positive definite matrices. These will now be treated as discrete observations from a continuous function of the PDS matrix $Y(P,t)$, i.e., $Y(P,t_k) = Y_k$. In $\mathbb{P}$, the Laplace–Beltrami operator $\mathcal{L}$ in the coordinates $P = (p_{ij})$ is given as

$$\mathcal{L} = \mathrm{tr}(P\partial_P)^2$$

where

$$\partial_P = \Big(\frac{1}{2}(1+\delta_{ij})\frac{\partial}{\partial p_{ij}}\Big)$$

and δ_{ij} is Kronecker's delta (Maass, 1955; James, 1968; Richards, 1985; Haff et al., 2011). We can compute the eigenfunctions in $\mathbb{P}$ by solving

$$\mathcal{L}\psi_k(P) = -\lambda_k\psi_k(P)$$

using the finite difference scheme (Chung and Taylor, 2004; Chung et al., 2015b). The eigenfunction of the Laplace–Beltrami (LB) operator involves zonal spherical functions (Richards, 1985, 2011). However, there are currently no available tools for computing these functions. Although the computation of LB eigenfunctions for large-scale PDS matrices can be fairly time consuming, we expect this to be solvable in desktop computers for most practical settings.

Since ψ_k spans $\mathbb{P}$, any linear combination of ψ_k will be again a PDS matrix. Thus, we can obtain the Fourier series expansion of $Y(P,t)$:

$$Y(P,t) = \sum_{k=0}^{K} y_k(t)\psi_k(P),$$

where

$$y_k(t) = \int_{\mathbb{P}^q} Y(P,t)\psi_k(P)\, dP$$

are the Fourier coefficients, which can be estimated using the least squares estimation (LSE), with the optimal truncation degree K determined by the forward model selection approach (Chung et al., 2007, 2008a).

The main advantage of using fixed basis over data-driven PCA-type basis is that only the coefficients of the basis will dynamically change and thus provide a far better interpretability of how objects evolve. From the LB basis, we can further develop a localized LB-wavelet basis (Tan and Qiu, 2015). Moreover, we can enforce sparsity using the principles of wavelet thresholding (Donoho et al., 1995). By adding a sparsity penalty, the method can more sharply identify subsets of basis functions that contain the most information on the dynamics, which we anticipate will provide better localization power.

13.2.3 PDE-Based Dynamic Models

We can further generalize on the manifold time series model by using the partial differential equation–based dynamic models that are able to accurately characterize more complex time-dependent structures. We propose to solve the following dynamic model

$$\frac{d}{dt}Y(P,t) = \mathcal{L}Y(P,t) + \epsilon(P),$$

where $\epsilon(P)$ is a zero mean noise. We can show that

$$Y(P,t) = \sum_{k=0}^{K} c_k e^{-\lambda_k t}\psi_k(P)$$

is the solution in the space spanned by up to K basis functions. The parameters c_k can be estimated via LSE. Since ψ_k spans $\mathbb{P}$, the solution is also in $\mathbb{P}$. The method can be used to estimate and characterize the evolution of brain connectivity over time. Although we expect LB-operator $\mathcal{L}$ to work, other more general differential operators might be more suitable for capturing the dynamic features of connectivity.

13.3 Persistent Homological Dynamic Network Model

In this section, we present a new simple but very effective *data-driven* approach to assess the dynamic pattern of resting state functional connectivity using recently popular persistent homology, an algebraic approach to quantifying topology (Lee et al., 2012). So far, persistent homology has been successfully applied to 3D static networks by building a graph filtration over changing edge weights, which results in 4D hypernetwork structure. For 4D dynamic network data that are changing over time, we propose to build a graph filtration over time and edge weights, making it a 5D hypernetwork structure.

A particular interest with respect to persistent homology is determining if the topological structure of the brain network at the resting state is changing or not. We can test this biological question using the topological invariants called Betti numbers β_j (Lee et al., 2018a). This is not an easy question and requires building a graph filtration over dynamically changing connectivity matrices.

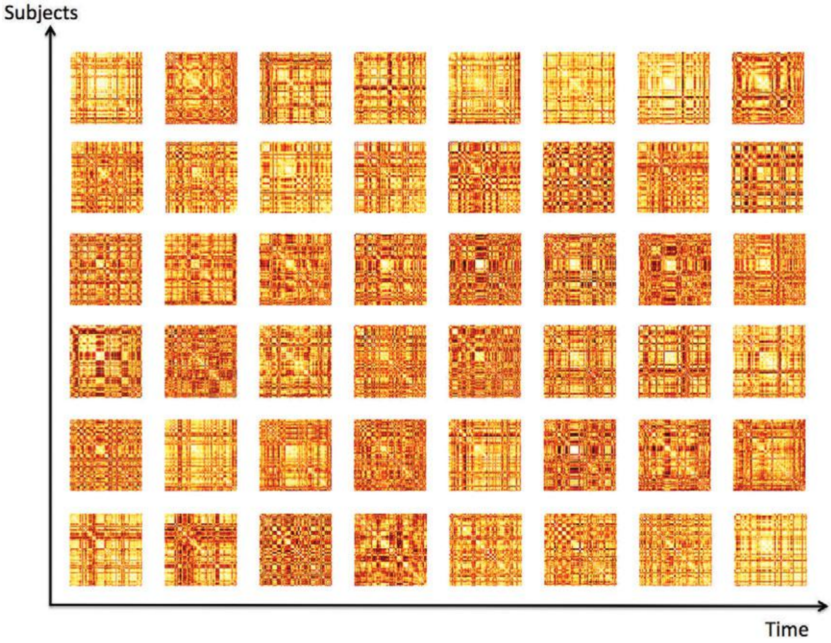


Figure 13.2 Dynamically changing resting-state connectivity represented using 116 × 116 correlation matrices. The figure shows six subjects over eight sliding windows obtained from HCP.

Assume we have a sequence of q connectivity matrices obtained through a sliding window method $G_1, G_2, \cdots, G_q$ (Allen et al., 2014) (Figure 13.2). We are interested in determining if

$$\begin{aligned} H_0 :&\ G_1 = G_2 = \cdots = G_q \\ &\ vs. \\ H_1 :&\ \text{ At least one } G_k \text{ is different.} \end{aligned}$$

This is a difficult multiple comparisons problem and requires additional data processing.

13.3.1 Multidimensional Graph Filtration

To make the above inference more stable, we build a persistent homological structure called the graph filtration, which is based on the nested graph structure (Chung et al., 2015a). It is not hard to build the nested graphs in dynamic networks (Figure 13.3).

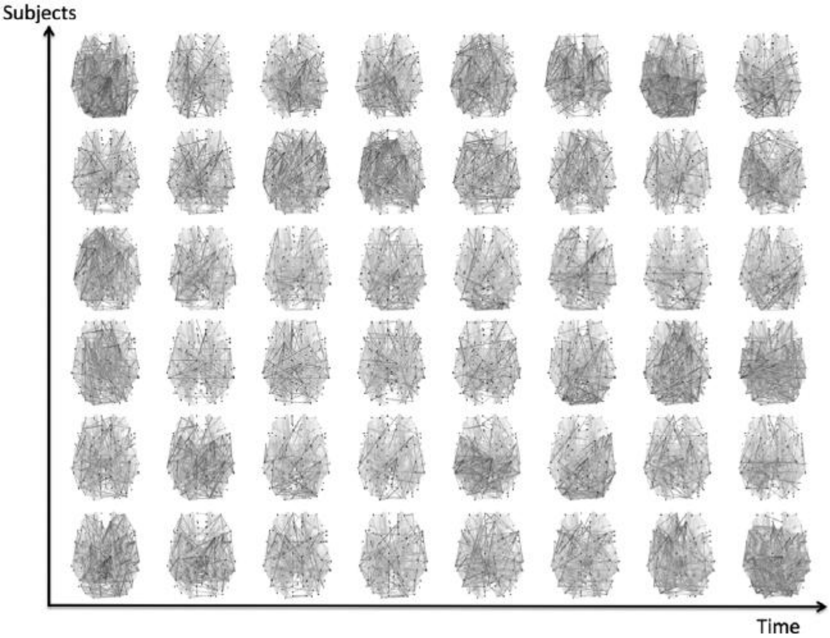


Figure 13.3 Dynamically changing resting-state connectivity represented using 3D graphs. The graphs are constructed by thresholding at correlation value 0.9 in Figure 13.2. The figure shows six subjects over eight sliding windows obtained from HCP.

Let

$$F_j = G_1 \cup G_2 \cup \cdots \cup G_j.$$

The union $\cup$ operation should be defined depending on applications. Then we have filtration over union:

$$F_1 \subset F_2 \subset \cdots \subset F_{q-1} \subset F_q. \tag{13.4}$$

Thus, testing H_0 vs. H_1 is equivalent to testing

$$\begin{aligned} H_0' :&\ F_1 = F_2 = \cdots = F_{q-1} = F_q. \\ &\ vs. \\ H_1' :&\ \text{At least one } F_k \text{ is different.} \end{aligned}$$

The graph filtration (13.4) requires multidimensional persistent homology for weighted graphs (Lee et al., 2017). This is done as follows. Threshold F_j at λ and obtain a binary graph filtration:

$$F_1|_\lambda \subset F_2|_\lambda \subset \cdots \subset F_q|_\lambda.$$

Further, we have

$$F_j|_{\lambda_1} \supset F_j|_{\lambda_2} \supset \cdots \supset F_j|_{\lambda_k}$$

for $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_k$.

Let $\beta_0(j,\lambda) = \beta_0(F_j|_\lambda)$ be the Betti-0 number. Then we have monotone sequences

$$\beta_0(1,\lambda) \ge \beta_0(2,\lambda) \ge \cdots \ge \beta_0(q,\lambda)$$

and

$$\beta_0(j,\lambda_1) \ge \beta_0(j,\lambda_2) \ge \cdots \ge \beta_0(j,\lambda_k).$$

Similarly, we also have

$$\beta_1(1,\lambda) \le \beta_1(2,\lambda) \le \cdots \le \beta_1(q,\lambda)$$

and

$$\beta_1(j,\lambda_1) \le \beta_1(j,\lambda_2) \le \cdots \le \beta_1(j,\lambda_k).$$

Hence $\beta_0(j,\lambda)$ and $\beta_1(j,\lambda)$ are 2D monotone functions. Then we test

$$\begin{aligned} &H_0 : \beta_i(1,\lambda) = \beta_i(2,\lambda) = \cdots = \beta_i(q,\lambda) \text{ for all } \lambda \\ &vs. \\ &H_1 : \text{ At least one } \beta_i(j,\lambda) \text{ is different for some } j. \end{aligned}$$

We are basically testing the equivalence of q monotone functions, which is a multiple comparisons problem and not necessarily an easy test. Usually 1D Betti-plots are shaped like logistic curves, which are S-shaped curves. So for each fixed j, we fit the logistic curve

$$f(\lambda) = \frac{p}{1 + e^{-k_j(\lambda - x_j)}},$$

where p is the total number of nodes, x_j is the midpoint, and k_j is the slop at the midpoint (Wang et al., 2016a). Then statistical inference will be done on the equivalence of vectors (x_j, k_j) instead of the equivalence of monotone curves. We can simply use existing multivariate data analysis techniques for this purpose (Flury, 1997).

Bibliography

Abdi, H. 2007. *The RV Coefficient and the Congruence Coefficient*. Sage.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**, 2274–2282.

Achard, S., and Bullmore, E. 2007. Efficiency and cost of economical brain functional networks. *PLoS Computational Biology*, **3**(2), e17.

Achard, S., Salvador, R., Whitcher, B., Suckling, J., and Bullmore, E.D. 2006. A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. *Journal of Neuroscience*, **26**, 63–72.

Adamic, L.A. 1999. The small world web. Pages 443–452 of: *Proceedings of ECDL*, vol. 99. Springer.

Adams, H., Tausz, A., and Vejdemo-Johansson, M. 2014. javaPlex: A research software package for persistent (Co) homology. Pages 129–136 of: *Mathematical Software–ICMS 2014*. Springer.

Adler, R.J. 1981. *The Geometry of Random Fields*. John Wiley & Sons.

Adler, R.J. 1990. *An Introduction to Continuity, Extrema, and Related Topics for General Gaussian Processes*. IMS.

Adler, R.J. 2000. On excursion sets, tube formulas and maxima of random fields. *Annals of Applied Probability*, **10**, 1–74.

Adler, R.J., and Taylor, J.E. 2007. *Random Fields and Geometry*. Springer Verlag.

Adler, R.J., Bobrowski, O., Borman, M.S., Subag, E., and Weinberger, S. 2010. Persistent homology for random fields and complexes. Pages 124–143 of: *Borrowing Strength: Theory Powering Applications: A Festschrift for Lawrence D. Brown*. Institute of Mathematical Statistics.

Allen, E.A., Damaraju, E., Plis, S.M., Erhardt, E.B., Eichele, T., and Calhoun, V.D. 2014. Tracking whole-brain connectivity dynamics in the resting state. *Cerebral Cortex*, **24**, 663–676.

Anderson, T.W. 1984. *An Introduction to Multivariate Statistical Analysis*. 2nd edn. Wiley.

Andrade, A., Kherif, F., Mangin, J., et al. J-B. 2001. Detection of fMRI activation using cortical surface mapping. *Human Brain Mapping*, **12**, 79–93.

Anirudh, R., Thiagarajan, J.J., Kim, I., and Polonik, W. 2016. Autism spectrum disorder classification using graph kernels on multidimensional time series. *arXiv preprint arXiv:1611.09897*.

Antoine, J.-P., Roşca, D., and Vandergheynst, P. 2010. Wavelet transform on manifolds: old and new approaches. *Applied and Computational Harmonic Analysis*, **28**, 189–202.

Antoine, J.B., Petra, A., and Max, A. 1996. Evaluation of ridge seeking operators for multimodality medical image matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**, 353–365.

Arai, K., and Barakbah, A.R. 2007. Hierarchical K-means: an algorithm for centroids initialization for K-means. *Reports of the Faculty of Science and Engineering*, **36**, 25–31.

Arfken, G.B. 2000. *Mathematical Methods for Physicists*. 5th edn. Academic Press.

Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. 2005. Fast and simple calculus on tensors in the Log-Euclidean framework. *Lecture Notes in Computer Science*, **3749**, 115–122.

Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. 2006. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, **56**, 411–421.

Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. 2007. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, **29**, 328–347.

Ashburner, J., and Friston, K. 2000. Voxel-based morphometry – the methods. *NeuroImage*, **11**, 805–821.

Ashburner, J, Good, C., and Friston, K.J. 2000. Tensor based morphometry. *NeuroImage*, **11S**, 465.

Avants, B.B., Cook, P.A., Ungar, L., Gee, J.C., and Grossman, M. 2010. Dementia induces correlated reductions in white matter integrity and cortical thickness: a multivariate neuroimaging study with sparse canonical correlation analysis. *NeuroImage*, **50**, 1004–1016.

Aylward, E.H., Minshew, N.J, Goldstein, G., et al. 1999. MRI volumes of amygdala and hippocampus in nonmentally retarded autistic adolescents and adults. *Neurology*, **53**, 2145–2150.

Babuška, I., Tempone, R., and Zouraris, G.E. 2004. Galerkin finite element approximations of stochastic elliptic partial differential equations. *Siam Journal of Numerical Analysis*, **42**, 800–825.

Bagarinao, E., Nakai, T., and Tanaka, Y. 2006. Real-time functional MRI: development and emerging applications. *Magnetic Resonance in Medical Sciences*, **5**, 157–165.

Banerjee, O., Ghaoui, L.E., d'Aspremont, A., and Natsoulis, G. 2006. Convex optimization techniques for fitting sparse Gaussian graphical models. Page 96 of: *Proceedings of the 23rd International Conference on Machine Learning*.

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. 2008. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, **9**, 485–516.

Banks, D., and Carley, K. 1994. Metric inference for social networks. *Journal of Classification*, **11**, 121–149.

Banuelos, R., and Burdzy, K. 1999. On the hot spots conjecture of J Rauch. *Journal of Functional Analysis*, **164**, 1–33.

Basser, P.J., and Pierpaoli, C. 1996. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI. *Journal of Magnetic Resonance, Series B*, **111**, 209–219.

Basser, P.J., Mattiello, J., and LeBihan, D. 1994. MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, **66**, 259–267.

Basser, P.J., Pajevic, S., Pierpaoli, C., Duda, J., and Aldroubi, A. 2000. In vivo tractography using DT-MRI data. *Magnetic Resonance in Medicine*, **44**, 625–632.

Bassett, D.S. 2006. Small-world brain networks. *The Neuroscientist*, **12**, 512–523.

Batchelor, P.G., Hill, D.L.G., Calamante, F., and Atkinson, D. 2001. Study of connectivity in the brain using the full diffusion tensor from MRI. *Lecture Notes in Computer Science*, **2082**, 121–133. Springer.

Batchelor, P.G., Calamante, F., Tournier, J.D., Atkinson, D., Hill, D.L., and Connelly, A. 2006. Quantification of the shape of fiber tracts. *Magnetic Resonance in Medicine*, **55**, 894–903.

Behrens, T.E.J., Berg, H.J., Jbabdi, S., Rushworth, M.F.S., and Woolrich, M.W. 2007. Probabilistic diffusion tractography with multiple fibre orientations: what can we gain? *NeuroImage*, **34**, 144–155.

Belkin, M., Niyogi, P., and Sindhwani, V. 2006. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, **7**, 2399–2434.

Benjamini, Y., and Hochberg, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of Royal Statistical Society, Series. B*, **57**, 289–300.

Benjamini, Y., and Yekutieli, D. 2001. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, **29**, 1165–1188.

Berge, A., Jensen, A.C., and Solberg, A.H.S. 2007. Sparse inverse covariance estimates for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, **45**, 1399.

Bernal-Rusiel, J., Atienza, M., and Cantero, J. 2008. Detection of focal changes in human cortical thickness: spherical wavelets versus Gaussian smoothing. *NeuroImage*, **41**, 1278–1292.

Betounes, D. 1998. *Partial Differential Equations for Computational Science: With Maple and Vector Analysis*. Springer.

Betzel, R.F., and Bassett, D.S. 2017. Multi-scale brain networks. *NeuroImage*, **160**, 73–83.

Bickel, P.J., and Levina, E. 2008. Regularized estimation of large covariance matrices. *Annals of Statistics*, **36**, 199–227.

Bien, J., and Tibshirani, R. 2011. Hierarchical clustering with prototypes via minimax linkage. *Journal of American Statistical Association*, **106**, 1075–1084.

Billingsley, P. 1995. *Convergence of Probability Measures*. 3rd edn. John Wiley & Sons.

Blakley, B.R., Quiñones, M.A., Crawford, M.S., and Jago, I. 1994. The validity of isometric strength tests. *Personnel Psychology*, **47**, 247–274.

Bocher, M. 1906. Introduction to the theory of Fourier's series. *Annals of Mathematics*, **7**, 81–152.

Böhm, W., and Hornik, K. 2010. A Kolmogorov–Smirnov test for r samples. *Institute for Statistics and Mathematics*, **Research Report Series**, Report 105.

Boothby, W.M. 1986. *An Introduction to Differential Manifolds and Riemannian Geometry*. 2nd edn. Academic Press.

Boubela, R.N., Kalcher, K., Huf, W., Našel, C., and Moser, E. 2016. Big data approaches for the analysis of large-scale fMRI data using Apache Spark and GPU processing: a demonstration on resting-state fMRI data from the Human Connectome Project. *Frontiers in Neuroscience*, **9**, 492.

Brezinski, C. 2004. Extrapolation algorithms for filtering series of functions, and treating the Gibbs phenomenon. *Numerical Algorithms*, **36**, 309–329.

Bronstein, M.M., and Kokkinos, I. 2010. Scale-invariant heat kernel signatures for non-rigid shape recognition. Pages 1704–1711 of: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE.

Bubenik, P. 2015. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, **16**, 77–102.

Bubenik, P., and Kim, P.T. 2007. A statistical approach to persistent homology. *Homology Homotopy and Applications*, **9**, 337–362.

Bullmore, E., and Sporns, O. 2009. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Review Neuroscience*, **10**, 186–198.

Bulow, T. 2004. Spherical diffusion for 3D surface smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 1650–1654.

Cachia, A., Mangin, J.-F., Riviére, D., et al. 2003a. A generic framework for parcellation of the cortical surface into gyri using geodesic Voronoï diagrams. *Image Analysis*, **7**, 403–416.

Cachia, A., Mangin, J.-F., Riviére, D., et al. 2003b. A primal sketch of the cortex mean curvature: a morphogenesis based approach to study the variability of the folding patterns. *IEEE Transactions on Medical Imaging*, **22**, 754–765.

Cao, J., and Worsley, K.J. 1999a. The detection of local shape changes via the geometry of Hotelling's T2 fields. *Annals of Statistics*, **27**, 925–942.

Cao, J., and Worsley, K.J. 1999b. The geometry of correlation fields with an application to functional connectivity of the brain. *Annals of Applied Probability*, **9**, 1021–1057.

Cao, J., and Worsley, K.J. 2001. Applications of random fields in human brain mapping. *Spatial Statistics: Methodological Aspects and Applications*, **159**, 170–182.

Carlsson, G., and Memoli, F. 2008. Persistent clustering and a theorem of J. Kleinberg. *arXiv preprint arXiv:0808.2241*.

Carlsson, G., and Mémoli, F. 2010. Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research*, **11**, 1425–1470.

Carlsson, G., Ishkhanov, T., De Silva, V., and Zomorodian, A. 2008. On the local behavior of spaces of natural images. *International Journal of Computer Vision*, **76**, 1–12.

Cassidy, B., Rae, C., and Solo, V. 2015. Brain activity: conditional dissimilarity and persistent homology. Pages 1356–1359 of: *IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, IEEE.

Castillo, R., Castillo, E., Guerra, R., Johnson, V.E., et al. 2009. A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets. *Physics in Medicine and Biology*, **54**, 1849–1870.

Catani, M., Howard, R.J., Pajevic, S., and Jones, D.K. 2002. Virtual in vivo interactive dissection of white matter fasciculi in the human brain. *NeuroImage*, **17**, 77–94.

Chaiken, S., and Kleitman, D.J. 1978. Matrix tree theorems. *Journal of Combinatorial Theory, Series A*, **24**, 377–381.

Chan, G., and Wood, A.T.A. 1997. Algorithm AS 312: an algorithm for simulating stationary Gaussian random fields. *Journal of the Royal Statistical Society: Series C*, **46**, 171–181.

Chan, T.F., Golub, G.H., and LeVeque, R.J. 1983. Algorithms for computing the sample variance: analysis and recommendations. *The American Statistician*, **37**, 242–247.

Chen, X., Zhang, H., Gao, Y., Wee, C.-Y., Li, G., and Shen, D. 2016. High-order resting-state functional connectivity network for MCI classification. *Human Brain Mapping*, **37**, 3282–3296.

Cheng, L., De, J., Zhang, X., Lin, F., and Li, H. 2014. Tracing retinal blood vessels by matrix-forest theorem of directed graphs. Pages 626–633 of: *International Conference on Medical Image Computing and Computer Assisted Intervention*. Springer.

Christakis, N.A., and Fowler, J.H. 2007. The spread of obesity in a large social network over 32 years. *New England Journal of Medicine*, **2007**, 370–379.

Chung, F.R.K., and Yau, S.T. 1997. Eigenvalue inequalities for graphs and convex subgraphs. *Communications in Analysis and Geometry*, **5**, 575–624.

Chung, M.K. 2001. *Statistical Morphometry in Neuroanatomy*. Ph.D. thesis, McGill University. www.stat.wisc.edu/~mchung/papers/thesis.pdf.

Chung, M.K. 2006. Heat kernel smoothing on unit sphere. Pages 467–473 of: *Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI)*, vol. I. IEEE.

Chung, M.K. 2007. *Correlation Coefficient*. Sage Publications.

Chung, M.K. 2012. *Computational Neuroanatomy: The Methods*. World Scientific.

Chung, M.K. 2013. *Statistical and Computational Methods in Brain Image Analysis*. CRC Press.

Chung, M.K. 2018. Statistical challenges of big brain network data. *Statistics and Probability Letter*, **136**, 79–82.

Chung, M.K., and Taylor, J. 2004. Diffusion smoothing on brain surface via finite element method. Pages 432–435 of: *Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI)*, vol. 1.

Chung, M.K., Worsley, K.J., Taylor, J., Ramsay, J., Robbins, S., and Evans, A.C. 2001a. Diffusion smoothing on the cortical surface. *NeuroImage*, **13**, S95.

Chung, M.K., Worsley, K.J., Paus, T., et al. 2001b. A unified statistical approach to deformation-based morphometry. *NeuroImage*, **14**, 595–606.

Chung, M.K., Worsley, K.J., Robbins, S., et al. 2003a. Deformation-based surface morphometry applied to gray matter deformation. *NeuroImage*, **18**, 198–213.

Chung, M.K., Lazar, M., Alexander, A.L., Lu, Y., and Davidson, R.J. 2003b. Probabilistic connectivity measure in diffusion tensor imaging via anisotropic kernel smoothing. *University of Wisconsin, Department of Statistics, Technical Report*, **1081**.

Chung, M.K., Worsley, K.J., Robbins, S., and Evans, A.C. 2003c. Tensor-based brain surface modeling and analysis. Pages 467–473 of: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. I.

Chung, M.K., Robbins, S., Dalton, K.M., Davidson, R.J., Alexander, A.L., and Evans, A.C. 2005a. Cortical thickness analysis in autism with heat kernel smoothing. *NeuroImage*, **25**, 1256–1265.

Chung, M.K., Robbins, S., and Evans, A.C. 2005b. Unified statistical approach to cortical thickness analysis. *Information Processing in Medical Imaging (IPMI), Lecture Notes in Computer Science*, **3565**, 627–638.

Chung, M.K., Dalton, K.M., Shen, L., Evans, A.C., and Davidson, R.J. 2007. Weighted Fourier representation and its application to quantifying the amount of gray matter. *IEEE Transactions on Medical Imaging*, **26**, 566–581.

Chung, M.K., Hartley, R., Dalton, K.M., and Davidson, R.J. 2008a. Encoding cortical surface by spherical harmonics. *Statistica Sinica*, **18**, 1269–1291.

Chung, M.K., Dalton, K.M., and Davidson, R.J. 2008b. Tensor-based cortical surface morphometry via weighted spherical harmonic representation. *IEEE Transactions on Medical Imaging*, **27**, 1143–1151.

Chung, M.K., Bubenik, P., and Kim, P.T. 2009a. Persistence diagrams of cortical surface data. *Proceedings of the 21st International Conference on Information Processing in Medical Imaging (IPMI), Lecture Notes in Computer Science (LNCS)*, **5636**, 386–397.

Chung, M.K., Singh, V., Kim, P.T., Dalton, K.M., and Davidson, R.J. 2009b. Topological characterization of signal in brain images using min–max diagrams. *MICCAI, Lecture Notes in Computer Science (LNCS)*, **5762**, 158–166.

Chung, M.K., Adluru, N., Lee, J.E., Lazar, M., Lainhart, J.E., and Alexander, A.L. 2010a. Cosine series representation of 3D curves and its application to white matter fiber bundles in diffusion tensor imaging. *Statistics and Its Interface*, **3**, 69–80. www.stat.wisc.edu/~mchung/papers/chung.2010.SII.pdf.

Chung, M.K., Worsley, K.J., Brendon, M.N., Dalton, K.M., and Davidson, R.J. 2010b. General multivariate linear modeling of surface shapes using SurfStat. *NeuroImage*, **53**, 491–505.

Chung, M.K., Hanson, J.L., Avants, B., Gee, J., Davidson, R.J., and Pollak, S.D. 2010c. Structural connectivity mapping via the tensor-based morphometry. Page 481 of: *16th Annual Meeting of the Organization for Human Brain Mapping*.

Chung, M.K., Hanson, J.L., Davidson, R.J., and Pollak, S.D. 2011a. Effect of family income on hippocampus growth: longitudinal study. Page 2697 of: *17th Annual Meeting of the Organization for Human Brain Mapping*.

Chung, M.K., Seo, S., Adluru, N., and Vorperian, H.K. 2011b. Hot spots conjecture and its application to modeling tubular structures. Pages 225–232 of: *International Workshop on Machine Learning in Medical Imaging*, vol. 7009.

Chung, M.K., Adluru, N., Dalton, K.M., Alexander, A.L., and Davidson, R.J. 2011c. Scalable brain network construction on white matter fibers. Page 79624G of: *Proceedings of SPIE*, vol. 7962.

Chung, M.K., Hanson, J.L., Lee, H., et al. 2013. Persistent homological sparse network approach to detecting white matter abnormality in maltreated children: MRI and DTI multimodal study. *MICCAI, Lecture Notes in Computer Science (LNCS)*, **8149**, 300–307.

Chung, M.K., Kim, S.-G., Schaefer, S.M., et al. 2014. Improved statistical power with a sparse shape model in detecting an aging effect in the hippocampus and amygdala. Page 90340Y of: *Medical Imaging 2014: Image Processing*, vol. 9034.

Chung, M.K., Hanson, J.L., Ye, J., Davidson, R.J., and Pollak, S.D. 2015a. Persistent homology in sparse regression and its application to brain morphometry. *IEEE Transactions on Medical Imaging*, **34**, 1928–1939.

Chung, M.K., Qiu, A., Seo, S., and Vorperian, H.K. 2015b. Unified heat kernel regression for diffusion, kernel smoothing and wavelets on manifolds and its application to mandible growth modeling in CT images. *Medical Image Analysis*, **22**, 63–76.

Chung, M.K., Vilalta-Gil, V., Lee, H., Rathouz, P.J., Lahey, B.B., and Zald, D.H. 2017a. Exact topological inference for paired brain networks via persistent homology. *Information Processing in Medical Imaging (IPMI), Lecture Notes in Computer Science (LNCS)*, **10265**, 299–310.

Chung, M.K., Hanson, J.L., Adluru, L., Alexander, A.L., Davidson, R.J., and Pollak, S.D. 2017b. Integrative structural brain network analysis in diffusion tensor imaging. *Brain Connectivity*, **7**, 331–346.

Chung, M.K., Chuang, Y.J., and Vorperian, H.K. 2017c. Online statistical inference for large-scale binary images. *MICCAI, Lecture Notes in Computer Science (LNCS)*, **10434**, 729–736.

Chung, M.K., Lee, H., Solo, V., Davidson, R.J., and Pollak, S.D. 2017d. Topological distances between brain networks. Pages 161–170 of: *International Workshop on Connectomics in Neuroimaging, Lecture Notes in Computer Science*.

Chung, M.K., Wang, Y., and Wu, G. 2018a. Heat kernel smoothing in irregular image domains. Pages 5101–5104 of: *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*.

Chung, M.K., Luo, Z., Adluru, A., Alexander, A.L., Richard, D.J., and Goldsmith, H.H. 2018b. Heritability of hierarchical structural brain network. Pages 554–557 of: *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*.

Clayden, J.D., Storkey, A.J., and Bastin, M.E. 2007. A probabilistic model-based approach to consistent white matter tract segmentation. *IEEE Transactions on Medical Imaging*, **11**, 1555–1561.

Cohen-Steiner, D., Edelsbrunner, H., and Harer, J. 2007. Stability of persistence diagrams. *Discrete and Computational Geometry*, **37**, 103–120.

Collins, D.L., Paus, T., Zijdenbos, A., et al. 1998. Age related changes in the shape of temporal and frontal lobes: an MRI study of children and adolescents. *Society of Neuroscience Abstracts*, **24**, 304.

Conturo, T.E., Lori, N.F., Cull, T.S., Akbudak, E., et al. 1999. Tracking neuronal fiber pathways in the living human brain. *National Academy of Sciences USA*, **96**, 10422–10427.

Conway, J.B. 1990. *A Course in Functional Analysis*. Springer.

Cook, P.A., Bai, Y., Nedjati-Gilani, S., et al. 2006. Camino: open-source diffusion–MRI reconstruction and processing. Page 2759 in: *14th Scientific Meeting of the International Society for Magnetic Resonance in Medicine*.

Cootes, T.F., Hill, A., Taylor, C.J., and Haslam, J. 1993. The use of active shape models for locating structures in medical images. *Lecture Notes in Computer Science*, **687**, 33–33.

Corey, D.M., Dunlap, W.P., and Burke, M.J. 1998. Averaging correlations: expected values and bias in combined Pearson rs and Fisher's z transformations. *Journal of General Psychology*, **125**, 245–261.

Corouge, I., Gouttard, S., and Gerig, G. 2004. Towards a shape model of white matter fiber bundles using diffusion tensor MRI. Pages 344–347 of: *IEEE International Symposium on Biomedical Imaging: Nano to Macro*.

Courant, R., and Hilbert, D. 1953. *Methods of Mathematical Physics*. English edn. Interscience.

Craddock, R.C., James, G.A., Holtzheimer, P.E., Hu, X.P., and Mayberg, H.S. 2012. A whole brain fMRI atlas generated via spatially constrained spectral clustering. *Human Brain Mapping*, **33**, 1914–1928.

David, O., Kiebel, S.J., Harrison, L.M., Mattout, J., Kilner, J.M., and Friston, K.J. 2006. Dynamic causal modeling of evoked responses in EEG and MEG. *NeuroImage*, **30**, 1255–1272.

de Goes, F., Goldenstein, S., and Velho, L. 2008. A hierarchical segmentation of articulated bodies. *Computer Graphics Forum*, **27**, 1349–1356.

de Silva, V., and Ghrist, R. 2007. Homological sensor networks. *Notices of the American Mathematical Society*, **54**, 10–17.

Deng, C.Y. 2011. A generalization of the Sherman–Morrison–Woodbury formula. *Applied Mathematics Letters*, **24**, 1561–1564.

Desikan, R.S., Ségonne, F., Fischl, B., et al. 2006. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *NeuroImage*, **31**, 968–980.

Ding, C., and He, X. 2004. K-means clustering via principal component analysis. Page 29 of: *Proceedings of the Twenty-First International Conference on Machine Learning*. ACM.

do Carmo, M.P. 1992. *Riemannian Geometry*. Prentice Hall, Inc.

Dobra, A., Hans, C., Jones, B., Nevins, J.R., Yao, G., and West, M. 2004. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, **90**, 196–212.

Dong, S., Bremer, P.T., Garland, M., Pascucci, V., and Hart, J.C. 2006. Spectral surface quadrangulation. Pages 1057–1066 of: *ACM SIGGRAPH 2006 Papers*. ACM.

Donoho, D.L., and Tsaig, Y. 2006. *Fast Solution of l_1-Norm Minimization Problems When the Solution May Be Sparse*. Citeseer.

Donoho, D.L., Johnstone, I.M., Kerkyacharian, G., and Picard, D. 1995. Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society. Series B (Methodological)*, **57**, 301–369.

Dougherty, E.R. 1999. *Random Processes for Image and Signal Processing*. IEEE Press.

Doyle, P.G., and Snell, J.L. 1984. *Random Walks and Electric Networks*. Mathematical Association of America.

Edelsbrunner, H., and Harer, J. 2008. Persistent homology: a survey. *Contemporary Mathematics*, **453**, 257–282.

Edelsbrunner, H., and Harer, J. 2010. *Computational Topology: An Introduction*. American Mathematical Society.

Edelsbrunner, H., Letscher, D., and Zomorodian, A. 2002. Topological persistence and simplification. *Discrete and Computational Geometry*, **28**, 511–533.

Efron, B. 1982. *The Jackknife, the Bootstrap and Other Resampling Plans*, vol. 38. SIAM.

Eguíluz, V.M., Chialvo, D.R., Cecchi, G.A., Baliki, M., and Apkarian, A.V. 2005. Scale-free brain functional networks. *Physical Review Letters*, **94**(1), 018102.

Embrechts, P., Resnick, S.I., and Samorodnitsky, G. 1999. Extreme value theory as a risk management tool. *North American Actuarial Journal*, **3**, 30–41.

Erdös, P., and Rényi, A. 1961. On the evolution of random graphs. *Bulletin of the International Statistical Institute*, **38**, 343–347.

Fan, J., Han, F., and Liu, H. 2014. Challenges of big data analysis. *National Science Review*, **1**, 293–314.

Fiedler, M. 1973. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, **23**, 298–305.

Finch, T. 2009. Incremental calculation of weighted mean and variance. *University of Cambridge*, **4**, 11–5.

Fischl, B., and Dale, A.M. 2000. Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proceedings of the National Academy of Sciences (PNAS)*, **97**, 11050–11055.

Fisher, R.A. 1915. Frequency distribution of the values of the correlation coefficient in samples of an indefinitely large population. *Biometrika*, **10**, 507–521.

Fisher, R.A. 1966. *The Design of Experiements*. 8 edn. Hafner.

Fletcher, P.T., Lu, C., Pizer, S.M., and Joshi, S. 2004. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, **23**, 995–1005.

Flury, B. 1997. *A First Course in Multivariate Statistics*. Springer.

Fornito, A., Zalesky, A., and Bullmore, E.T. 2010. Network scaling effects in graph analytic studies of human resting-state fMRI data. *Frontiers in Systems Neuroscience*, **4**, 1–16.

Fornito, A., Zalesky, A., and Bullmore, E. 2016. *Fundamentals of Brain Network Analysis*. Academic Press.

Foster, J., and Richards, F.B. 1991. The Gibbs phenomenon for piecewise-linear approximation. *American Mathematical Monthly*, **98**.

Fox, J. 2002. *An R and S-Plus Companion to Applied Regression*. Sage Publications, Inc.

Fox, J., Friendly, M., and Monette, G. 2009. Visualizing hypothesis tests in multivariate linear models: the heplots package for R. *Computational Statistics*, **24**, 233–246.

Frangi, A.F., Niessen, W.J., Vincken, K.L., and Viergever, M.A. 1998. Multiscale vessel enhancement filtering. Pages 130–137 of: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 1496.

Freeman, L.C. 1977. A set of measures of centrality based on betweenness. *Sociometry*, **40**, 35–41.

Friedman, J., Hastie, T., and Tibshirani, R. 2008. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**, 432.

Frigge, M., Hoaglin, D.C., and Iglewicz, B. 1989. Some implementations of the boxplot. *American Statistician*, 50–54.

Friston, K.J. 1994. Functional and effective connectivity in neuroimaging: A synthesis. *Human Brain Mapping*, **2**, 56–78.

Friston., K.J. 2002. *A short history of statistical parametric mapping in functional neuroimaging*. Technical report. Wellcome Department of Imaging Neuroscience, ION, UCL.

Friston, K.J., Frith, C.D., Liddle, P.F., and Frackowiak, R.S.J. 1993a. Functional connectivity: the principal-component analysis of large (PET) data sets. *Journal of Cerebral Blood Flow and Metabolism*, **13**, 5–14.

Friston, K.J., Frith, C.D., and Frackowiak, R.S.J. 1993b. Time-dependent changes in effective connectivity measured with PET. *Human Brain Mapping*, **1**, 69–79.

Friston, K.J., Holmes, A.P., Worsley, K.J., Poline, J.-P., Frith, C.D., and Frackowiak, R.S.J. 1995. Statistical parametric maps in functional imaging: A general linear approach. *Human Brain Mapping*, **2**, 189–210.

Friston, K.J., Harrison, L., and Penny, W. 2003. Dynamic causal modelling. *NeuroImage*, **19**, 1273–1302.

Fukunaga, K., and Koontz, W.L.G. 1970. Application of the Karhunen–Loeve expansion to feature selection and ordering. *IEEE Transactions on Computers*, **100**, 311–318.

Gaser, C., Volz, H.-P., Kiebel, S., Riehemann, S., and Sauer, H. 1999. Detecting structural changes in whole brain based on nonlinear deformations: application to schizophrenia research. *NeuroImage*, **10**, 107–113.

Genovese, C.R., Lazar, N.A., and Nichols, T. 2002. Thresholding of statistical maps in functional neuroimaging using the false discovery rate. *NeuroImage*, **15**, 870–878.

Gerig, G., Styner, M., Jones, D., Weinberger, D., and Lieberman, J. 2001. Shape analysis of brain ventricles using SPHARM. Pages 171–178 of: *MMBIA*.

Ghrist, R. 2008. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, **45**, 61–75.

Gibbons, J. D., and Chakraborti, S. 2011. *Nonparametric Statistical Inference*. Chapman & Hall/CRC Press.

Gibbs, J.W. 1898. Fourier's series. *Nature*, **59**, 200.

Girvan, M., and Newman, M.E.J. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, **99**, 7821.

Gladwell, G.M.L., and Zhu, H. 2002. Courant's nodal line theorem and its discrete counterparts. *Quarterly Journal of Mechanics and Applied Mathematics*, **55**, 1–15.

Goldsmith, J., Crainiceanu, C.M., Caffo, B.S., and Reich, D.S. 2011. Penalized functional regression analysis of white-matter tract profiles in multiple sclerosis. *NeuroImage*, **57**, 431–439.

Goldsmith, J., Crainiceanu, C.M., Caffo, B., and Reich, D. 2012. Longitudinal penalized functional regression for cognitive outcomes on neuronal tract measurements. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **61** 453–469.

Gong, G., He, Y., Concha, L., et al. 2009. Mapping anatomical connectivity patterns of human cerebral cortex using in vivo diffusion tensor imaging tractography. *Cerebral Cortex*, **19**, 524–536.

Gottlieb, D., and Shu, C.-W. 1997. On the Gibbs phenomenon and its resolution. *SIAM Review*, **39**, 644–668.

Gower, J. C., and Ross, G. J. S. 1969. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society, Series C (Applied statistics)*, **18**, 54–64.

Greicius, M.D., Srivastava, G., Reiss, A.L., and Menon, V. 2004. Default-mode network activity distinguishes Alzheimer's disease from healthy aging: evidence from functional MRI. *Proceedings of the National Academy of Sciences*, **101**, 4637–4642.

Grigoryan, A. 1999. Analytic and geometric background of recurrence and non-explosion of the Brownian motion on Riemannian manifolds. *Bulletin of the American Mathematical Society*, **36**, 135–249.

Gritsenko, A., Lindquist, M.A., Kirk, G.R., and Chung, M.K. 2018. Hill climbing optimized twin classification using resting-state functional MRI. *arXiv*, 1807.00244.

Gritsenko, A, Kirk G.R., and Chung, M.K. 2018. Resting-state fMRI segmentation in spatio-temporal domain using supervoxels. Page 2595, of: *Organization for Human Brain Meeting Annual Meeting*.

Gruen, A., and Akca, D. 2005. Least squares 3D surface and curve matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, **59**, 151–174.

Gu, X., Wang, Y.L., Chan, T.F., Thompson, T.M., and Yau, S.T. 2004. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging*, **23**, 1–10.

Gueziec, A., Pennec, X., and Ayache, N. 1997. Medical image registration using geometeric hashing. *IEEE Computational Science and Engineering*, **4**, 29–41.

Haff, L.R., Kim, P.T., Koo, J.-Y., and Richards, D.S.P. 2011. Minimax estimation for mixtures of Wishart distributions. *Annals of Statistics*, **39**, 3417–3440.

Hagmann, P., Thiran, J.P., Vandergheynst, P., Clarke, S., Meuli, R., and Lausanne, S. 2000. Statistical fiber tracking on DT-MRI data as a potential tool for morphological brain studies. Page 216 in: *ISMRM Workshop on Diffusion MRI: Biophysical Issues*.

Hagmann, P., Kurant, M., Gigandet, X., et al. 2007. Mapping human whole-brain structural networks with diffusion MRI. *PLoS One*, **2**(7), e597.

Hagmann, P., Cammoun, L., Gigandet, X., et al. 2008. Mapping the structural core of human cerebral cortex. *PLoS Biolology*, **6**(7), e159.

Hair, J.F., Tatham, R.L, Anderson, R.E., and Black, W.C. 1998. *Multivariate Data Analysis*. Prentice Hall, Inc.

Hall, K.M. 1970. An r-dimensional quadratic placement algorithm. *Management Science*, **17**, 219–229.

Ham, J., Lee, D.D., Mika, S., and Schölkopf, B. 2004. A kernel view of the dimensionality reduction of manifolds. Page 47 of: *Proceedings of the Twenty-First International Conference on Machine Learning*.

Ham, J., Lee, D., and Saul, L. 2005. Semisupervised alignment of manifolds. Pages 120–127 of: *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, vol. 10.

Hammond, D.K., Vandergheynst, P., and Gribonval, R. 2011. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, **30**, 129–150.

Han, X., Jovicich, J., Salat, D., et al. 2006. Reliability of MRI-derived measurements of human cerebral cortical thickness: the effects of field strength, scanner upgrade and manufacturer. *NeuroImage*, **32**, 180–194.

Hanson, J.L., Adluru, N., Chung, M.K., Alexander, A.L., Davidson, R.J., and Pollak, S.D. 2013. Early neglect is associated with alterations in white matter integrity and cognitive functioning. *Child Development*, **84**, 1566–1578.

Hart, J.C. 1999. Computational topology for shape modeling. Pages 36–43 of: *Proceedings of the International Conference on Shape Modeling and Applications*.

Harville, D.A. 1997. *Matrix Algebra from a Statistician's Perspective*. Springer Verlag.

Hayasaka, S., Peiffer, A.M., Hugenschmidt, C.E., and Laurienti, P.J. 2007. Power and sample size calculation for neuroimaging studies by non-central random field theory. *NeuroImage*, **37**, 721–730.

He, Y., Chen, Z.J., and Evans, A.C. 2007. Small-world anatomical networks in the human brain revealed by cortical thickness from MRI. *Cerebral Cortex*, **17**, 2407–2419.

He, Y., Chen, Z., and Evans, A. 2008. Structural insights into aberrant topological patterns of large-scale cortical networks in Alzheimer's disease. *Journal of Neuroscience*, **28**, 4756.

Heo, G., Gamble, J., and Kim, P.T. 2012. Topological analysis of variance and the maxillary complex. *Journal of the American Statistical Association*, **107**, 477–492.

Histed, M.H., Bonin V., and Reid, R.C. 2009. Direct activation of sparse, distributed populations of cortical neurons by electrical microstimulation. *Neuron*, **63**, 508–522.

Holzrichter, M., and Oliveira, S. 1999. A graph based method for generating the Fiedler vector of irregular problems. *Parallel and Distributed Processing, Lecture Notes in Computer Science (LNCS)*, **1586**, 978–985.

Horak, D., Maletić, S., and Rajković, M. 2009. Persistent homology of complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, **2009**, P03034.

Hsieh, C.-J., Sustik, M.A., Dhillon, I.S., Ravikumar, P.K., and Poldrack, R. 2013. BIG & QUIC: sparse inverse covariance estimation for a million variables. Pages 3165–3173 of: *Advances in Neural Information Processing Systems*.

Huang, S., Li, J., Sun, L., et al. 2009. Learning brain connectivity of Alzheimer's disease from neuroimaging data. Pages 808–816 of: *Advances in Neural Information Processing Systems*.

Huang, S., Li, J., Sun, L., et al. 2010. Learning brain connectivity of Alzheimer's disease by sparse inverse covariance estimation. *NeuroImage*, **50**, 935–949.

Hutchinson, J.E. 1981. Fractals and self-similarity. *Indiana University Mathematics Journal*, **30**, 713–747.

James, A.T. 1968. Calculation of zonal polynomial coefficients by use of the Laplace–Beltrami operator. *Annals of Mathematical Statistics*, **39**, 1711–1718.

Jbabdi, S., Bellec, P., Toro, R., Daunizeau, J., Pélégrini-Issac, M., and Benali, H. 2008. Accurate anisotropic fast marching for diffusion-based geodesic tractography. *International Journal of Biomedical Imaging*, **2008**, 1–12.

Jerri, A.J. 1998. *The Gibbs Phenomenon in Fourier Analysis, Splines and Wavelet Approximations*. Springer.

Jezzard, P., and Clare, S. 1999. Sources of distortion in functional MRI data. *Human Brain Mapping*, **8**, 80–85.

Jolliffe, I.T., Trendafilov, N.T., and Uddin, M. 2003. A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics*, **12**, 531–547.

Jones, S.E., Buchbinder, B.R., and Aharon, I. 2000. Three-dimensional mapping of cortical thickness using Laplace's equation. *Human Brain Mapping*, **11**, 12–32.

Joshi, A.A., Shattuck, D. W., Thompson, P. M., and Leahy, R. M. 2009. A parameterization-based numerical method for isotropic and anisotropic diffusion smoothing on non-flat Surfaces. *IEEE Transactions on Image Processing*, **18**, 1358–1365.

Joshi, S.C. 1998. *Large Deformation Diffeomorphisms and Gaussian Random Fields for Statistical Characterization of Brain Sub-Manifolds*. Ph.D. thesis, Washington University, St. Louis.

Joshi, S.C., Wang, J., Miller, M.I., Van Essen, D.C., and Grenander, U. 1995. Differential geometry of the cortical surface. Pages 304–311 of: *Vision Geometry IV, Vol. 2573, Proceedings of the SPIE 1995 International Symposium on Optical Science, Engineering and Instrumentation*.

Joshi, S.C., Davis, B., Jomier, M., and Gerig, G. 2004. Unbiased diffeomorphic atlas construction for computational anatomy. *NeuroImage*, **23**, 151–160.

Kazi-Aoual, F., Hitier, S., Sabatier, R., and Lebreton, J.D. 1995. Refined approximations to permutation tests for multivariate inference. *Computational Statistics and Data Analysis*, **20**, 643–656.

Khalid, A., Kim, B.S., Chung, M.K., Ye, J.C., and Jeon, D. 2014. Tracing the evolution of multi-scale functional networks in a mouse model of depression using persistent brain network homology. *NeuroImage*, **101**, 351–363.

Kiebel, S.J., Poline, J.-P., Friston, K.J., Holmes, A.P., and Worsley, K.J. 1999. Robust smoothness estimation in statistical parametric maps using standardized residuals from the general linear model. *NeuroImage*, **10**, 756–766.

Kim, S.-G., Chung, M.K., Hanson, J.L., et al. 2011. Structural connectivity via the tensor-based morphometry. Pages 808–811 of: *The Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI)*.

Kim, S.-G., Lee, H.K., Chung, M.K., et al. 2012a. Agreement between the white matter connectivity based on the tensor-based morphometry and the volumetric white matter parcellations based on diffusion tensor imaging. Pages 42–45 of: *The Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI)*.

Kim, W.H., Pachauri, D., Hatt, C., Chung, M.K., Johnson, S., and Singh, V. 2012b. Wavelet based multi-scale shape features on arbitrary surfaces for cortical thickness discrimination. Pages 1250–1258 of: *Advances in Neural Information Processing Systems*.

Kim, W.H., Adluru, N., Chung, M.K., et al. 2015. Multi-resolution statistical analysis of brain connectivity graphs in preclinical Alzheimer's disease. *NeuroImage*, **118**, 103–117.

Kirby, M. 2000. *Geometric Data Analysis: An Empirical Approach to Dimensionality Reduction and the Study of Patterns*. John Wiley & Sons, Inc.

Kishon, E., Hastie, T., and Wolfson, H. 1990. 3D curve matching using splines. Pages 589–591 of: *Proceedings of the European Conference on Computer Vision*.

Knuth, D. 1981. *The Art of Computing, Volume 2: Seminumerical Algorithms*. Addison-Wesley.

Koch, M.A., Norris, D.G., and Hund-Georgiadis, M. 2002. An investigation of functional and anatomical connectivity using magnetic resonance imaging. *NeuroImage*, **16**, 241–250.

Kondor, R., Howard, A., and Jebara, T. 2007. Multi-object tracking with representations of the symmetric group. Page 5 of: *International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 1.

Koren, Y., and Harel, D. 2002. A multi-scale algorithm for the linear arrangement problem. **2573**, 296–309.

Korfiatis, P.D., Kalogeropoulou, C., Karahaliou, A.N., Kazantzi, A.D., and Costaridou, L.I. 2011. Vessel tree segmentation in presence of interstitial lung disease in MDCT. *IEEE Transactions on Information Technology in Biomedicine*, **15**, 214–220.

Kreyszig, E. 1959. *Differential Geometry*. University of Toronto Press.

Kwapien, S., and Woyczynski, W.A. 1992. *Random Series and Stochastic Integrals: Single and Multiple*. Birkhauser.

Lazar, M., Weinstein, D.M., Tsuruda, J.S., et al. 2003. White matter tractography using tensor deflection. *Human Brain Mapping*, **18**, 306–321.

Lee, H., Chung, M.K., Kang, H., and Lee, D.S. 2014. 2014. Hole detection in metabolic connectivity of Alzheimer's disease using k-Laplacian. Pages 297–304 of: *International Conference on Medical Image Computing and Computer-Assisted Intervention, Lecture Notes in Computer Science*.

Lee, H., Chung, M.K., Kang, H., Kim, B.-N., and Lee, D.S. 2011a. Computing the shape of brain networks using graph filtration and Gromov–Hausdorff metric. *MICCAI, Lecture Notes in Computer Science*, **6892**, 302–309.

Lee, H., Chung, M.K., Kang, H., Kim, B.-N., and Lee, D.S. 2011b. Discriminative persistent homology of brain networks. Pages 841–844 of: *IEEE International Symposium on Biomedical Imaging (ISBI)*.

Lee, H., Lee, D.S.., Kang, H., Kim, B.-N., and Chung, M.K. 2011c. Sparse brain network recovery under compressed sensing. *IEEE Transactions on Medical Imaging*, **30**, 1154–1165.

Lee, H., Kang, H., Chung, M.K., Kim, B.-N., and Lee, D.S. 2012. Persistent brain network homology from the perspective of dendrogram. *IEEE Transactions on Medical Imaging*, **31**, 2267–2277.

Lee, H., Kang, H., Chung, M.K., Lim, S., Kim, B.-N., and Lee, D.S. 2017. Integrated multimodal network approach to PET and MRI based on multidimensional persistent homology. *Human Brain Mapping*, **38**, 1387–1402.

Lee, H., Chung, M.K., Kang, H., Choi, H., Kim, Y.K., and Lee, D.S. 2018a. Abnormal hole detection in brain connectivity by kernel density of persistence diagram and Hodge Laplacian. Pages 20–23 of: *IEEE International Symposium on Biomedical Imaging (ISBI)*.

Lee, M.-H., Kim, D.-Y., Chung, M.K., Alexander, A.L., and Davidson, R.J. 2018b. Topological properties of the brain network constructed using the epsilon-neighbor method. *IEEE Transactions on Biomedical Engineering*, **65**, 2323–2333.

Leemans, A., Sijbers, J., Backer, S. De, Vandervliet, E., and Parizel, P. 2006. Multiscale white matter fiber tract coregistration: a new feature-based approach to align diffusion tensor data. *Magnetic Resonance in Medicine*, **55**, 1414–1423.

Lehoucq, R.B., and Sorensen, D.C. 1996. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications*, **17**, 789–821.

Lepore, N., Brun, C.A., Chiang, M.C., et al. 2006. Multivariate statistics of the Jacobian matrices in tensor based morphometry and their application to HIV/AIDS. *Lecture Notes in Computer Science*, **4190**, 191–198.

Lerch, J.P., Worsley, K., Shaw, W.P., et al. 2006. Mapping anatomical correlations across cerebral cortex (MACACC) using cortical thickness from MRI. *NeuroImage*, **31**, 993–1003.

Lévy, B. 2006. Laplace–Beltrami eigenfunctions towards an algorithm that "understands" geometry. Page 13 of: *IEEE International Conference on Shape Modeling and Applications*.

Li, Y., Liu, Y., Li, J., et al. 2009. Brain anatomical network and intelligence. *PLoS Computational Biology*, **5**(5), e1000395.

Lindvere, L., Janik, R., Dorr, A., et al. 2013. Cerebral microvascular network geometry changes in response to functional stimulation. *Neuroimage*, **71**, 248–259.

Luscombe, N.M., Babu, M.M., Yu, H., Snyder, M., Teichmann, S.A., and Gerstein, M. 2004. Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, **431**, 308–312.

Maass, H. 1955. Die Bestimmung der Dirichletreihen mit Grössencharakteren zu den Modulformen n-ten Grades. *Journal of Indian Mathematical Society*, **19**, 1–23.

MacDonald, J.D., Kabani, N., Avis, D., and Evans, A.C. 2000. Automated 3-D extraction of inner and outer surfaces of cerebral cortex from MRI. *NeuroImage*, **12**, 340–356.

Mahadevan, S., and Maggioni, M. 2006. Value function approximation with diffusion wavelets and Laplacian eigenfunctions. *Advances in Neural Information Processing Systems*, **18**, 843.

Malladi, R, and Ravve, I. 2002. Fast difference schemes for edge enhancing Beltrami flow. Pages 343–357 of: *Proceedings of Computer Vision–ECCV, Lecture Notes in Computer Science (LNCS)*, vol. 2350.

Mandelbrot, B.B. 1982. *The Fractal Geometry of Nature*. Freeman.

Marrelec, G., Krainik, A., Duffau, H., et al. 2006. Partial correlation for functional brain interactivity investigation in functional MRI. *NeuroImage*, **32**, 228–237.

Marrelec, G., Kim, J., Doyon, J., and Horwitz, B. 2009. Large-scale neural model validation of partial correlation analysis for effective connectivity investigation in functional MRI. *Human Brain Mapping*, **30**, 941–950.

Marsden, J.E., and Hughes, T.J.R. 1983. *Mathematical Foundations of Elasticity*. Dover Publications, Inc.

Mather, M., Canli, T., English, T., et al. 2004. Amygdala responses to emotionally valanced stimuli in older and younger adults. *Psychological Science*, **15**, 259–263.

Mazumder, R., and Hastie, T. 2012. Exact covariance thresholding into connected components for large-scale graphical LASSO. *Journal of Machine Learning Research*, **13**, 781–794.

McIntosh, A.R., and Lobaugh, N.J. 2004. Partial least squares analysis of neuroimaging data: applications and advances. *NeuroImage*, **23**, 250–263.

McIntosh, A.R., Bookstein, F.L., Haxby, J.V., and Grady, C.L. 1996. Spatial pattern analysis of functional brain images using partial least squares. *NeuroImage*, **3**, 143–157.

Mclntosh, A.R., and Gonzalez-Lima, F. 1994. Structural equation modeling and its application to network analysis in functional brain imaging. *Human Brain Mapping*, **2**, 2–22.

Mémoli, F. 2008. Gromov–Hausdorff distances in Euclidean Spaces. Pages 1–8 of: *Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (CVPR Workshop, NORDIA'08).*

Miller, M.I., Banerjee, A., Christensen, G.E., et al. 1997. Statistical methods in computational anatomy. *Statistical Methods in Medical Research*, **6**, 267–299.

Milliken, J.K., and Edland, S.D. 2000. Mixed effect models of longitudinal Alzheimer's disease data: a cautionary note. *Statistics in Medicine*, **19**, 1617–1629.

Milnor, J. 1973. *Morse Theory*. Princeton University Press.

Moakher, M. 2005. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, **26**, 735–747.

Moakher, M., and Zéraï, M. 2011. The Riemannian geometry of the space of positive-definite matrices and its application to the regularization of positive-definite matrix-valued data. *Journal of Mathematical Imaging and Vision*, **40**, 171–187.

Mori, S., and van Zijl, P.C.M. 2002. Fiber tracking: principles and strategies – a technical review. *NMR in Biomedicine*, **15**, 468–480.

Mori, S., Crain, B.J., Chacko, V.P., and van Zijl, P.C. 1999. Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging. *Annals of Neurology*, **45**, 256–269.

Mori, S., Oishi, K., Jiang, H., et al. 2008. Stereotaxic white matter atlas based on diffusion tensor imaging in an ICBM template. *NeuroImage*, **40**, 570–582.

Morozov, D. 2008. *Homological Illusions of Persistence and Stability*. Ph.D. thesis, Duke University.

Naiman, D.Q. 1990. Volumes for tubular neighborhoods of spherical polyhedra and statistical inference. *Annals of Statistics*, **18**, 685–716.

Nain, D., Styner, M., Niethammer, M., et al. 2007. Statistical shape analysis of brain structures using spherical wavelets. Pages 209–212 in: *IEEE Symposium on Biomedical Imaging ISBI.*

Newman, M.E.J. 2003. The structure and function of complex networks. *SIAM Review*, **45**, 167.

Newman, M.E.J., and Watts, D.J. 1999. Scaling and percolation in the small-world network model. *Physical Review E*, **60**, 7332–7342.

Newman, M.E.J., Strogatz, S.H., and Watts, D.J. 2001. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, **64**, 26118.

Newman, M.E.J., Barabasi, A.L., and Watts, D.J. 2006. *The Structure and Dynamics of Networks*. Princeton University Press.

Neykov, M., Lu, J., and Liu, H. 2016. Combinatorial inference for graphical models. *arXiv preprint arXiv:1608.03045*.

Nichols, T., and Hayasaka, S. 2003. Controlling the familywise error rate in functional neuroimaging: a comparative review. *Statistical Methods in Medical Research*, **12**, 419–446.

Nichols, T.E., and Holmes, A.P. 2002. Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human Brain Mapping*, **15**, 1–25.

Nilsson, J., Sha, F., and Jordan, M.I. 2007. Regression on manifolds using kernel dimension reduction. Pages 697–704 of: *Proceedings of the 24th International Conference on Machine Learning*. ACM.

O'Donnell, L.J., and Westin, C.F. 2007. Automatic tractography segmentation using a high-dimensional white matter atlas. *IEEE Transactions on Medical Imaging*, **26**, 1562–1575.

O'Donnell, L.J., Kubicki, M., Shenton, M.E., Dreusicke, M.H., Grimson, W.E., and Westin, C.F. 2006. A method for clustering white matter fiber tracts. *American Journal of Neuroradiology*, **27**, 1032–1036.

Øksendal, B. 2010. *Stochastic Differential Equations: An Introduction with Applications*. Springer.

Ombao, H., and Van Bellegem, S. 2008. Evolutionary coherence of nonstationary signals. *IEEE Transactions on Signal Processing*, **56**, 2259–2266.

Ombao, H., Von Sachs, R., and Guo, W. 2005. SLEX analysis of multivariate nonstationary time series. *Journal of the American Statistical Association*, **100**, 519–531.

Ombao, H., Fiecas, M., Ting, C.-M., and Low, Y. F. 2017. Statistical models for brain signals with properties that evolve across trials. *NeuroImage*, **180**, 609–618.

Osborne, M.R., Presnell, B., and Turlach, B.A. 2000. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, **20**, 389–404.

Osher, S., and Fedkiw, R.P. 2003. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag.

Palande, S., Jose, V., Zielinski, B., Anderson, J., Fletcher, P.T., and Wang, B. 2017. Revisiting abnormalities in brain network architecture underlying autism using topology-inspired statistical inference. Pages 98–107 of: *International Workshop on Connectomics in Neuroimaging*.

Parker, G.J.M., Wheeler-Kingshott, C.A.M., and Barker, G.J. 2002. Estimating distributed anatomical connectivity using fast marching methods and diffusion tensor imaging. *IEEE Transactions on Medical Imaging*, **21**, 505–512.

Paul, W., and Baschnagel, J. 1999. *Stochastic Processes from Physics to Finance*. Berlin: Springer Verlag.

Peng, J., Wang, P., Zhou, N., and Zhu, J. 2009. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, **104**, 735–746.

Penny, W.D., Stephan, K.E., Mechelli, A., and Friston, K.J. 2004. Comparing dynamic causal models. *NeuroImage*, **22**, 1157–1172.

Pepe, A., Auzias, G., De Guio, F., et al. 2015. Spectral clustering based parcellation of fetal brain MRI. Pages 152–155 of: *IEEE International Symposium on Biomedical Imaging (ISBI)*.

Perona, P., and Malik, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 629–639.

Persoon, E., and Fu, K.S. 1977. Shape discrimination using Fourier descriptors. *IEEE Transactions on Systems, Man and Cybernetics*, **7**, 170–179.

Petri, G., Expert, P., Turkheimer, F., et al. 2014. Homological scaffolds of brain functional networks. *Journal of the Royal Society Interface*, **11**, 20140873.

Pinehiro, J.C., and Bates, D.M. 2002. *Mixed Effects Models in S and S-Plus*. 3rd edn. Springer.

Plumbley, M.D. 2005. Geometry and homotopy for l_1 sparse representations. *Proceedings of SPARS*, **5**, 206–213.

Pothen, A., and Fan, C.J. 1990. Computing the block triangular form of a sparse matrix. *ACM Transactions on Mathematical Software (TOMS)*, **16**, 324.

Prabhu, P., and Anbazhagan, N. 2011. Improving the performance of k-means clustering for high dimensional data set. *International Journal on Computer Science and Engineering*, **3**, 2317–2322.

Qiu, A., Bitouk, D., and Miller, M.I. 2006. Smooth functional and structural maps on the neocortex via orthonormal bases of the Laplace–Beltrami operator. *IEEE Transactions on Medical Imaging*, **25**, 1296–1396.

Qiu, A., Lee, A., Tan, M., and Chung, M.K. 2015. Manifold learning on brain functional networks in aging. *Medical Image Analysis*, **20**, 52–60.

Raftery, A.E., Newton, M.A., Satagopan, J.M., and Krivitsky, P.N. 2006. Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. *Memorial Sloan Kettering Cancer Center, Dept. of Epidemiology and Biostatistics Working Paper Series. Working Paper 6.*

Ramsay, J.O., and Silverman, B.W. 1997. *Functional Data Analysis*. Springer Verlag.

Reid, M., and Szendròi, B. 2005. *Geometry and Topology*. Cambridge University Press.

Reuter, M. 2010. Hierarchical shape segmentation and registration via topological features of Laplace–Beltrami eigenfunctions. *International Journal of Computer Vision*, **89**, 287–308.

Reuter, M., Wolter, F.-E., Shenton, M., and Niethammer, M. 2009. Laplace–Beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis. *Computer-Aided Design*, **41**, 739–755.

Richards, D.S.P. 1985. Applications of invariant differential operators to multivariate distribution theory. *SIAM Journal on Applied Mathematics*, **45**, 280–288.

Richards, D.S.P. 2011. High-dimensional random matrices from the classical matrix groups, and generalized hypergeometric functions of matrix argument. *Symmetry*, **3**, 600–610.

Robert, P., and Escoufier, Y. 1976. A unifying tool for linear multivariate statistical methods: the RV-coefficient. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **25**, 257–265.

Roy, S.N. 1953. On a heuristic method of test construction and its use in multivariate analysis. *Annals of Mathematics and Statistics*, **24**, 220–238.

Rubinov, M., and Sporns, O. 2010. Complex network measures of brain connectivity: uses and interpretations. *NeuroImage*, **52**, 1059–1069.

Rubinov, M., Knock, S. A., Stam, C. J., et al. 2009. Small-world properties of nonlinear brain activity in schizophrenia. *Human Brain Mapping*, **30**, 403–416.

Salmond, CH, Ashburner, J., Vargha-Khadem, F., Connelly, A., Gadian, DG, and Friston, KJ. 2002. Distributional assumptions in voxel-based morphometry. *NeuroImage*, **17**, 1027–1030.

Salvador, R., Suckling, J., Coleman, M. R., Pickard, J. D., Menon, D., and Bullmore, E. 2005. Neurophysiological architecture of functional magnetic resonance images of human brain. *Cerebral Cortex*, **15**, 1332–1342.

Sato, Y., Nakajima, S., Shiraga, N., et al. 1998. Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, **2**, 143–168.

Schäfer, J., and Strimmer, K. 2005. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, **4**, 32.

Schmidt, V., and Spodarev, E. 2005. Joint estimators for the specific intrinsic volumes of stationary random sets. *Stochastic Processes and Their Applications*, **115**, 959–981.

Schölkopf, B., and Smola, A.J. 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

Seo, S., Chung, M.K., and Vorperian, H.K. 2010. Heat kernel smoothing using Laplace–Beltrami eigenfunctions. Pages 505–512 of: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*. Lecture Notes in Computer Science, vol. 6363.

Shang, Y., Deklerck, R., Nyssen, E., et al. 2011. Vascular active contour for vessel tree segmentation. *IEEE Transactions on Biomedical Engineering*, **58**, 1023–1032.

Shattuck, D.W., Mirza, M., Adisetiyo, V., et al. 2008. Construction of a 3D probabilistic atlas of human cortical structures. *NeuroImage*, **39**, 1064–1080.

Shawe-Taylor, J., and Cristianini, N. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Shen, L., and Chung, M.K. 2006. Large-scale modeling of parametric surfaces using spherical harmonics. Pages 294–301 in: *Third International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*.

Shen, L., Ford, J., Makedon, F., and Saykin, A. 2004. Surface-based approach for classification of 3D neuroanatomical structures. *Intelligent Data Analysis*, **8**, 519–542.

Shen, X., Papademetris, X., and Constable, R.T. 2010. Graph-theory based parcellation of functional subunits in the brain from resting-state fMRI data. *NeuroImage*, **50**, 1027–1035.

Shinkareva, S.V., Ombao, H.C., Sutton, B.P., Mohanty, A., and Miller, G.A. 2006. Classification of functional brain images with a spatio-temporal dissimilarity map. *NeuroImage*, **33**, 63–71.

Siegmund, D.O., and Worsley, K.J. 1996. Testing for a signal with unknown location and scale in a stationary Gaussian random field. *Annals of Statistics*, **23**, 608–639.

Silver, N.C., and Hollingsworth, S.C. 1989. A FORTRAN 77 program for averaging correlation coefficients. *Behavior Research Methods*, **21**, 647–650.

Smirnov, N.V. 1939. Estimate of deviation between empirical distribution functions in two independent samples. *Bulletin of Moscow University*, **2**, 3–16.

Smith, R.L. 1989. Extreme value analysis of environmental time series: an application to trend detection in ground-level ozone. *Statistical Science*, **4**, 367–377.

Sochen, N., Kimmel, R., and Malladi, R. 1998. A general framework for low level vision. *IEEE Transactions on Image Processing*, **7**, 310–318.

Song, C., Havlin, S., and Makse, H.A. 2005. Self-similarity of complex networks. *Nature*, **433**, 392–395.

Sporns, O., and Zwi, J.D. 2004. The small world of the cerebral cortex. *Neuroinformatics*, **2**, 145–162.

Sporns, O., Tononi, G., and Edelman, GM. 2000. Theoretical neuroanatomy: relating anatomical and functional connectivity in graphs and cortical connection matrices. *Cerebral Cortex*, **10**, 127.

Staempfli, P., Jaermann, T., Crelier, G.R., Kollias, S., Valavanis, A., and Boesiger, P. 2006. Resolving fiber crossing using advanced fast marching tractography based on diffusion tensor imaging. *NeuroImage*, **30**, 110–120.

Staib, L.H., and Duncan, J.S. 1992. Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**, 1061–1075.

Stam, C.J., Tewarie, P., Van Dellen, E., van Straaten, E.C.W., and Hillebrand, A. 2014. The trees and the forest: characterization of complex brain networks with minimum spanning trees. *International Journal of Psychophysiology*, **92**, 129–138.

Steinke, F., and Hein, M. 2008. Non-parametric regression between manifolds. *Advances in Neural Information Processing Systems*, **21**, 1561–1568.

Stevens, C.F. 1995. *The Six Core Theories of Modern Physics*. MIT Press.

Stroup, W.W. 2012. *Generalized Linear Mixed Models: Modern Concepts, Methods and Applications*. CRC Press.

Subasi, A., and Ercelebi, E. 2005. Classification of EEG signals using neural network and logistic regression. *Computer Methods and Programs in Biomedicine*, **78**, 87–99.

Sun, J., Ovsjanikov, M., and Guibas, L. J. 2009. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, **28**, 1383–1392.

Supekar, K., Menon, V., Rubin, D., Musen, M., and Greicius, M.D. 2008. Network analysis of intrinsic functional brain connectivity in Alzheimer's disease. *PLoS Computational Biology*, **4**(6), e1000100.

Talairach, J., and Tournoux, P. 1988. *Co-Planar Stereotactic Atlas of the Human Brain: 3-Dimensional Proportional system: An Approach to Cerebral Imaging*. Thieme, Stuttgart.

Tan, M., and Qiu, A. 2015. Spectral Laplace–Beltrami wavelets with applications in medical images. *IEEE Transactions on Medical Imaging*, **34**, 1005–1017.

Tang, B., Sapiro, G., and Caselles, V. 1999. Direction diffusion. Pages 2:1245–1252 of: *The Proceedings of the Seventh IEEE International Conference on Computer Vision*.

Taubin, G. 2000. Geometric signal processing on polygonal meshes. *EUROGRAPHICS*.

Taylor, J.E., and Worsley, K.J. 2007. Detecting sparse signals in random fields, with an application to brain mapping. *Journal of the American Statistical Association*, **102**, 913–928.

Taylor, J.E., and Worsley, K.J. 2008. Random fields of multivariate test statistics, with applications to shape analysis. *Annals of Statistics*, **36**, 1–27.

Taylor, P.N., Wang, Y., and Kaiser, M. 2017. Within brain area tractography suggests local modularity using high resolution connectomics. *Scientific Reports*, **7**, 39859.

Tench, C.R., Morgan, P.S., Blumhardt, L.D., and Constantinescu, C. 2002. Improved white matter fiber tracking using stochastic labeling. *Magnetic Resonance in Medicine*, **48**, 677–683.

Tenenbaum, J.B., De Silva, V., and Langford, J.C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, **290**, 2319–2323.

Thirion, B., Varoquaux, G., Dohmatob, E., and Poline, J.-B. 2014. Which fMRI clustering gives good brain parcellations? *Frontiers in Neuroscience*, **8**, 167.

Thompson, P.M., MacDonald, D., Mega, M.S., Holmes, C.J., Evans, A.C., and Toga, A.W. 1997. Detection and mapping of abnormal brain structure with a probabilistic atlas of cortical surfaces. *Journal of Computer Assisted Tomography*, **21**, 567–581.

Thompson, P.M., Cannon, T.D., Narr, K.L., et al. 2001. Genetic influences on brain structure. *Nature Neuroscience*, **4**, 1253–1258.

Thottakara, P., Lazar, M., Johnson, S.C., and Alexander, A.L. 2006. Probabilistic connectivity and segmentation of white matter using tractography and cortical templates. *NeuroImage*, **29**, 868–878.

Tibshirani, R. 1996. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, **58**, 267–288.

Timm, N.H., and Mieczkowski, T.A. 1997. *Univariate and Multivariate General Linear Models: Theory and Applications Using SAS Software*. SAS Publishing.

Ting, C.-M., Ombao, H., Samdin, S.B., and Salleh, S.-H. 2018. Estimating dynamic connectivity states in fMRI using regime-switching factor models. *IEEE Transactions on Medical Imaging*, **37**, 1011–1023.

Tlusty, T. 2007. A relation between the multiplicity of the second eigenvalue of a graph Laplacian, Courants nodal line theorem and the substantial dimension of tight polyhedral surfaces. *Electronic Journal of Linear Algebra*, **16**, 315–24.

Topaz, C.M., Ziegelmeier, L., and Halverson, T. 2015. Topological data analysis of biological aggregation models. *PLoS One*, e0126383.

Tuzhilin, A.A. 2016. Who invented the Gromov–Hausdorff distance? *arXiv preprint arXiv:1612.00728*.

Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., et al. 2002. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *NeuroImage*, **15**, 273–289.

Uddin, L.Q., Kelly, A.M.C., Biswal, B.B., et al. 2008. Network homogeneity reveals decreased integrity of default-mode network in ADHD. *Journal of Neuroscience Methods*, **169**, 249–254.

Valdés-Sosa, P.A., Sánchez-Bornot, J.M., Lage-Castellanos, A., et al. 2005. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society B: Biological Sciences*, **360**, 969–981.

Vallet, B., and Lévy, B. 2008. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum*, **27**, 251–260.

Van Dijk, K.R.A., Sabuncu, M.R., and Buckner, R.L. 2012. The influence of head motion on intrinsic functional connectivity MRI. *NeuroImage*, **59**, 431–438.

Van Essen, D.C., Ugurbil, K., Auerbach, E., et al. 2012. The Human Connectome Project: a data acquisition perspective. *NeuroImage*, **62**, 2222–2231.

Vilanova, A., Berenschot, G., and van Pul, C. 2004. DTI visualization with stream-surfaces and evenly-spaced volume seeding. Pages 173–182 of: *VisSym04 Joint Eurographics-IEEETCVG Symposium on Visualization, Conference Proceedings*.

Wang, Y., Kang, J., Kemmer, P.B., and Guo, Y. 2016a. An efficient and reliable statistical method for estimating functional connectivity in large scale brain networks using partial correlation. *Frontiers in Neuroscience*, **10**, 123.

Wang, Y., Ting, C.-M., and Ombao, H. 2016b. Modeling effective connectivity in high-dimensional cortical source signals. *IEEE Journal of Selected Topics in Signal Processing*, **10**, 1315–1325.

Wang, Y., Chung, M.K., Dentico, D., Lutz, A., and Davidson, R.J. 2017. Topological network analysis of electroencephalographic power maps. Pages 134–142 of: *International Workshop on Connectomics in NeuroImaging, Lecture Notes in Computer Science (LNCS)*, vol. 10511.

Wang, Y., Ombao, H., and Chung, M.K. 2018. Topological data analysis of single-trial electroencephalographic signals. *Annals of Applied Statistics*, **12**, 1506–1534.

Watts, D.J., and Strogatz, S.H. 1998. Collective dynamics of small-world networks. *Nature*, **393**(6684), 440–442.

Wijk, B. C. M., Stam, C. J., and Daffertshofer, A. 2010. Comparing brain networks of different size and connectivity density using graph theory. *PloS One*, **5**, e13701.

Wilbraham, H. 1848. On a certain periodic function. *Cambridge and Dublin Mathematical Journal*, **3**, 198–201.

Wilk, M.B., and Gnanadesikan, R. 1968. Probability plotting methods for the analysis of data. *Biometrika*, **55**, 1.

Witten, D.M. Friedman, J.H., and Simon, N. 2011. New insights and faster computations for the graphical LASSO. *Journal of Computational and Graphical Statistics*, **20**, 892–900.

Wong, E., Palande, S., Wang, B., Zielinski, B., Anderson, J., and Fletcher, P.T. 2016. Kernel partial least squares regression for relating functional brain network topology to clinical measures of behavior. Pages 1303–1306 of: *IEEE International Symposium on Biomedical Imaging (ISBI)*.

Worsley, K.J. 1994. Local maxima and the expected Euler characteristic of excursion sets of χ^2, F and t fields. *Advances in Applied Probability*, **26**, 13–42.

Worsley, K.J. 2003. Detecting activation in fMRI data. *Statistical Methods in Medical Research.*, **12**, 401–418.

Worsley, K.J., Evans, A.C., Marrett, S., and Neelin, P. 1992. A three-dimensional statistical analysis for CBF activation studies in human brain. *Journal of Cerebral Blood Flow and Metabolism*, **12**, 900–918.

Worsley, K.J., Poline, J-B., Vandal, A.C., and Friston, K.J. 1995. Test for distributed, non-focal brain activations. *NeuroImage*, **2**, 173–181.

Worsley, K.J., Marrett, S., Neelin, P., and Evans, A.C. 1996a. Searching scale space for activation in PET images. *Human Brain Mapping*, **4**, 74–90.

Worsley, K.J., Marrett, S., Neelin, P., Vandal, A.C., Friston, K.J., and Evans, A.C. 1996b. A unified statistical approach for determining significant signals in images of cerebral activation. *Human Brain Mapping*, **4**, 58–73.

Worsley, K.J., Cao, J., Paus, T., Petrides, M., and Evans, A.C. 1998. Applications of random field theory to functional connectivity. *Human Brain Mapping*, **6**, 364–7.

Worsley, K.J., Taylor, J.E., Tomaiuolo, F., and Lerch, J. 2004. Unified univariate and multivariate random field theory. *NeuroImage*, **23**, S189–S195.

Worsley, K.J., Chen, J.I., Lerch, J., and Evans, A.C. 2005a. Comparing functional connectivity via thresholding correlations and singular value decomposition. *Philosophical Transactions of the Royal Society B: Biological Sciences*, **360**, 913.

Worsley, K.J., Charil, A., Lerch, J., and Evans, A.C. 2005b. Connectivity of anatomical and functional MRI data. Pages 1534–1541 of: *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*, vol. 3.

Worsley, K.J., Taylor, J.E., Carbonell, F., et al. 2009. SurfStat: a MATLAB toolbox for the statistical analysis of univariate and multivariate surface and volumetric data using linear mixed effects models and random field theory. *NeuroImage*, **47**, S102.

Wu, G., Wang, Q., Lian, J., and Shen, D. 2013. Estimating the 4D respiratory lung motion by spatiotemporal registration and super-resolution image reconstruction. *Medical Physics*, **40**(3), 031710.

Wu, Z., and Leahy, R. 1993. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 1101–1113.

Xavier, J., and Manton, J.H. 2006. On the generalization of AR processes to Riemannian manifolds. Pages V–V of: *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 5.

Xin, B., Hu, L., Wang, Y., and Gao, W. 2015. Stable feature selection from brain sMRI. Pages 1910–1916 in: *Proceedings of the Twenty-Nineth AAAI Conference on Artificial Intelligence*.

Yaglom, A.M. 1987. *Correlation Theory of Stationary and Related Random Functions Vol. I: Basic Results*. Springer Verlag.

Yger, F., and Rakotomamonjy, A. 2011. Wavelet kernel learning. *Pattern Recognition*, **44**(10-11), 2614–2629.

Yoo, K., Lee, P., Chung, M.K., Sohn, W.S., et al. 2017. Degree-based statistic and center persistency for brain connectivity analysis. *Human Brain Mapping*, **38**, 165–181.

Yoruk, E., Acar, B., and Bammer, R. 2005. A physical model for DT-MRI based connectivity map computation. *Lecture Notes in Computer Science*, **3749**, 213.

Young, M.P. 1992. Objective analysis of the topological organization of the primate cortical visual system. *Nature*, **358**, 152–155.

Yushkevich, P.A., Zhang, H., Simon, T.J., and Gee, J.C. 2007. Structure-specific statistical mapping of white matter tracts using the continuous medial representation. Pages 1–8 of: *IEEE 11th International Conference on Computer Vision (ICCV)*.

Zalesky, A., and Fornito, A. 2009. A DTI-derived measure of cortico-cortical connectivity. *IEEE Transactions on Medical Imaging*, **27**, 1023–1036.

Zalesky, A., Fornito, A., Harding, I.H., et al. 2010. Whole-brain anatomical networks: does the choice of nodes matter? *NeuroImage*, **50**, 970–983.

Zhang, B., Hsu, M., and Dayal, U. 1999. K-harmonic means-a data clustering algorithm. *Hewlett-Packard Labs Technical Report HPL-1999-124*.

Zhang, B.-T. 2008. Hypernetworks: a molecular evolutionary architecture for cognitive learning and memory. *IEEE Computational Intelligence Magazine*, **3**, 49–63.

Zhang, H., Yushkevich, P.A., Alexander, D.C., and Gee, J.C. 2006. Deformable registration of diffusion tensor MR images with explicit orientation optimization. *Medical Image Analysis*, **10**, 764–785.

Zhang, H., Avants, B.B., Yushkevich, P.A., et al. 2007a. High-dimensional spatial normalization of diffusion tensor images improves the detection of white matter differences: an example study using amyotrophic lateral sclerosis. *IEEE Transactions on Medical Imaging*, **26**, 1585–1597.

Zhang, H., van Kaick, O., and Dyer, R. 2007b. Spectral methods for mesh processing and analysis. Pages 1–22 of: *EUROGRAPHICS*.

Zhang, H., van Kaick, O., and Dyer, R. 2010. Spectral mesh processing. *Computer Graphics Forum*, **29**, 1865–1894.

Zhang, H., Chen, X., Shi, F., et al. 2016. Topographical information-based high-order functional connectivity and its application in abnormality detection for mild cognitive impairment. *Journal of Alzheimer's Disease*, **54**, 1095–1112.

Zhao, F., Zhang, H., Rekik, I., An, Z., and Shen, D. 2018. Diagnosis of autism spectrum disorders using multi-level high-order functional networks derived from resting-state functional MRI. *Frontiers in Human Neuroscience*, 184.

Zhu, X., Suk, H.-I., and Shen, D. 2014. Matrix-similarity based loss function and feature selection for Alzheimer's disease diagnosis. Pages 3089–3096 of: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zipunnikov, V., Caffo, B., Yousem, D. M., Davatzikos, C., Schwartz, B.S., and Crainiceanu, C. 2011a. Functional principal components model for high-dimensional brain imaging. *NeuroImage*, **58**, 772–784.

Zipunnikov, V., Greven, S., Caffo, B., Reich, D.S., and Crainiceanu, C. 2011b. Longitudinal high-dimensional data analysis. *Johns Hopkins University, Dept. of Biostatistics Working Papers*.

Zomorodian, A.J. 2001. *Computing and Comprehending Topology: Persistence and Hierarchical Morse Complexes*. Urbana–Champaign: Ph.D. thesis, University of Illinois.

Zomorodian, A.J. 2009. *Topology for Computing*. Cambridge Monographs on Applied and Computational Mathematics, vol. 16. Cambridge University Press.

Zomorodian, A.J., and Carlsson, G. 2005. Computing persistent homology. *Discrete and Computational Geometry*, **33**, 249–274.

Zou, H., Hastie, T., and Tibshirani, R. 2006. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, **15**, 265–286.

Index