Estimation of Gaussian Networks and Modular Brain Functional Networks

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Chen Gao

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Wei Pan, Ph.D.

November, 2017

Acknowledgements

I am grateful to my advisor, Professor Wei Pan, for his guidance and encouragement.

Abstract

Graphical models are intuitive tools to demonstrate dependence relation between variables of interest. Both undirected and directed graphical models are widely used in many applications, such as reconstructing gene expression/co-expression networks and brain functional networks. A popular model for undirected graphs is the Gaussian graphical model, where conditional independence can be inferred from the absence of an edge in the graph. Another approach for estimating undirected graphs does not depend on the distribution of the data. Instead, the resulting network is constructed through transformation of empirical sample correlation and node connectivity. The estimated network connectivity measures can be used as a secondary phenotype for association tests with genotypes. Finally, new methods have been proposed to estimate directed Gaussian graphs. The direction of an edge allows easier interpretation of causal relation between nodes in the graph.

We first aim to estimate multiple Gaussian graphs in the presence of sample heterogeneity, where the independent samples may come from different and unknown populations or distributions. We embed in the framework of a Gaussian mixture model one of two recently proposed methods for estimating multiple precision matrices in Gaussian graphical models. Secondly, we adapt a weighted gene co-expression network analysis (WGCNA) framework to resting-state fMRI (rs-fMRI) data to identify modular structures in brain functional networks. We propose applying a new adaptive test built on the proportional odds model (POM) that can be applied to a high-dimensional setting, where the number of variables (p) can exceed the sample size (n) in addition to the usual $p < n$ setting. Finally, we implemented a new method for estimating directed acyclic graph (DAG) as an R package, and demonstrated its use via application to a real data set and simulation studies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Graphical models are widely used to illustrate dependence relation between variables. A typical graph G is composed of V the set of nodes and E the set of edges. An edge in E indicates dependence between two nodes that are connected by the edge. The nodes and edges in a graphical model also forms a network. In this work, network and graph are used interchangeably. The application of graphical models includes gene expression networks [2, 3], brain functional networks [4, 5], social networks [6] and so on. This thesis focuses on statistical methods related to estimation and statistical testing of graphs.

The Gaussian graphical model, which is based on multivariate normal distribution (also called the Gaussian distribution), is an example of the graphical models. A non-zero off-diagonal element Σ_{ij}^{-1} in the inverse covariance matrix Σ^{-1} corresponds to independence of the two variables i, j, conditional on all other variable [7]. Estimation of a Gaussian graphical model is equivalent to estimation of the inverse covariance matrix in the Gaussian distribution. However, when the dimension of variables p exceeds the number of samples n in the data set, the empirical covariance matrix of the Gaussian distribution cannot be inverted. Estimation of the Gaussian graphical model in high-dimensional setting has become an interesting topic. A variety of methods use penalty terms to constrain the parameter space of the inverse covariance matrix, and successfully transforms the task to a constrained optimization problem [8, 2]. While these methods for estimating Gaussian graphs are popular, there is a need to estimate multiple graphs with commonnalities as suggested by real-world evidence [1]. Recently,

new methods have proposed to address the problem of estimating multiple Gaussian graphs [9, 3]. These new methods can be adapted to estimate multiple networks in the presence of sample heterogeneity, where the independent samples (i.e. observations) may come from different and unknown populations or distributions. In low-dimensional setting, the parameters in a Gaussian mixture model can be obtained via the EM algorithm [10]. In high-dimensional setting, penalization is needed for parameter estimation [11, 12]. By recognizing the relation between Gaussian graphical models and Gaussian mixture models, we embed in the EM algorithm one of two recently proposed methods for estimating multiple precision matrices in Gaussian graphical models. We demonstrate the feasibility and potential usefulness of the proposed methods in an application to glioblastoma subtype discovery and differential gene network analysis with a microarray gene expression data set. Our method is able to achieve simultaneous discovery of unknown disease subtypes and detection of differential gene (dys)regulations in functional genomics. We also conduct realistic simulation studies to evaluate and compare the performance of various methods.

Graphical models can be applied to construct brain functional networks[13, 14]. Disruption of connectivity in the brain functional network is related to many pathological conditions in the brain, such as Alzheimer's disease, Schizophrenia [15, 16]or autism [17]. A brain functional network may consist of records of more than 100 regions of interst (ROIs) [18]. Due to its high dimensionality and high noise levels, analysis of a large brain functional network may not be powerful. While some investigators restricted their analysis a smaller network with fewer ROIs [18], another line of approaches aims to decompose a large network into smaller subcomponents called modules. Although several methods exist for estimating brain functional networks, such as the sample correlation matrix or graphical lasso for a sparse precision matrix [19], it is still difficult to extract modules from such network estimates. Motivated by these considerations, we adapt a weighted gene co-expression network analysis (WGCNA) framework [20] to resting-state fMRI (rs-fMRI) data to identify modular structures in brain functional networks. Modular structures are identified by using topological overlap matrix (TOM) elements in hierarchical clustering. We propose applying a new adaptive test built on the proportional odds model (POM) that can be applied to a high-dimensional setting [21], where the number of variables (p) can exceed the sample size (n) in addition to

the usual $p < n$ setting. We applied our proposed methods to the ADNI data to test for associations between a genetic variant of the APOE4 allele and either the whole brain functional network or its various subcomponents using various connectivity measures. We uncovered several modules based on the control cohort, and some of them were marginally associated with the APOE4 variant and several other SNPs.

The graphical models discussed in Chapter 2 and 3 estimate undirected graphs, and the edges in undirected graphs do not have directions. The direction of dependence cannot be inferred from the edges in the undirected graphs. In contrast, directed graphs consist of directed edges between nodes. The directed edges have intuitive interpretation of causal relation of the nodes in a graph. Of particular interest is the directed acyclic graphs (DAGs) models, which exclude directed circles in the graph. Several methods have been proposed for estimating DAGs. For the low-dimensional setting where the number of nodes is relatively small, a few methods have been proposed by either reducing the search space [22] or greedy search[23]. However, these methods struggles in the high-dimensional setting because the search space grows super exponentially in the number of nodes [24]. Another class of methods originated from the PC algorithm [25], which uses conditional independence relationship to delete recursive edges from an undirected graph. Kalisch and Bühlmann [25] adapted the PC algorithm for estimating the skeleton of DAGs in the high-dimensional setting. Recently, Yuan et al. developed a new approach for estimation of DAGs in highdimensional setting. In this article, we first introduced the method for estimation of directed acyclic Gaussian graph, and implemented this method as the R package gDAG. We also reviewed the graphical lasso and its extension as competing methods for estimating graphs. We compared the performance of gDAG with graphical lasso and other competing methods. We also demonstrated an application of the gDAG package to the gene expression and SNP data set by Webster et al. [26].

Chapter 2 describes a new method for estimating Gaussian graphs in Gaussian mixture model setting. Chapter 3 applies a weighted network model for the estimation of brain functional networks. Adaptive association test is applied to test for the association between network connectivity measures and genotypes. Chapter 4 describes an R package that implements a new method for estimating directed acyclic graphs.

Chapter 2

Estimation of multiple networks in Gaussian mixture models

2.1 Introduction

We consider the problem of estimating multiple networks in the presence of sample heterogeneity; that is, the samples come from several populations with different Gaussian distributions, however it is unknown which samples are from which distributions. The precision matrix of each distribution corresponds to a network. This is related to but differs from the usual task of inferring and contrasting multiple networks in Gaussian graphical models, where it is known which samples are from which distributions (Guo et al. 2011; Danaher et al. 2014; Zhu et al. 2014 [27, 3, 9]). Although Gaussian mixture models are widely used for model-based clustering, our primary goal is for estimation and comparison of cluster-specific precision matrices, for which existing model-based clustering methods (McLachlan and Peel 2001; Fraley and Raftery 2006; Zhou et al. 2009 [28, 29, 12]) are not suitable. The existing model-based clustering methods either specify a common precision matrix or estimate multiple unconstrained cluster-specific precision matrices; due to the lack of a fusion penalty or other mechanisms, the cluster-specific precision matrix estimates are either exactly the same or completely different. On the other hand, in many applications one would expect both commonalities and differences among the cluster-specific precision matrices. Accounting for their commonalities not only improves statistical estimation efficiency through

information borrowing, but also enhances the ability of interpretation with a focus on few possible changes across the cluster-specific precision matrices.

Our proposed methods were motivated by genomic applications to disease subtype discovery while accounting for differential gene expression and/or differential gene regulations across (unknown) disease subtypes. This is in contrast to existing methods allowing for only differential gene expression in disease subtype discovery (Verhaak et al. 2010 [1]). Arguably, a biologically more interesting problem is not only in detecting differential gene expression, but also in discovering gene dysregulations, across to-be-discovered disease subtypes, which will facilitate understanding disease mechanisms and thus developing individualized treatments.

Our approach is in the framework of multivariate Gaussian mixture modeling (McLachlan and Peel 2001 [28]). The majority of the existing literature on mixture modeling focus on regularizing only the mean parameters with diagonal covariance matrices (Pan and Shen 2007; Wang and Zhu 2008; Xie et al. 2008 [11, 30, 31]), though some (Zhou et al. 2009; Hill and Mukerjee 2013; Wu et al. 2013 [12, 32, 33]) have started considering regularization of the covariance parameters too, all of which, however, do not touch on the key issue of identifying both common and varying substructures of the precision matrices across the components of a mixture model. Since these methods always give different networks for different populations unless a common network is assumed, they do not address the question of interest here: which parts of the networks change with the populations. To address this question, we propose embedding one of the current methods of estimating multiple Gaussian graphical models (Danaher et al. 2014; Zhu et al. 2014 [3, 9]) in the EM algorithm (Dempster et al. 1977 [10]) for the Gaussian mixture model, for which existing algorithms can be effectively used in the M-step of an EM algorithm for a Gaussian mixture model.

Since these methods apply a fusion penalty to shrink multiple networks towards each other, they not only are statistically more efficient with information borrowing, but also facilitate interpretation in identifying differential network substructures. In particular, due to the use of a non-convex penalty, the method of Zhu et al. (2014) [9] strives to uncover the commonalities among multiple networks while maintaining their unique substructures too.

Due to the connections to and differences from our current problem, we briefly review the literature on Gaussian graphical models *without sample heterogeneity*; that is, it is known that the samples come from the same Gaussian distribution. Gaussian graphical models are commonly used to describe conditional dependence relationships between interacting variables for continuous multivariate data. They are widely applied to reveal the structures in gene regulatory networks ([34, 35]), protein interaction networks ([2, 36, 37]) and brain functional connectivity ([38, 37]). Each network or graph consists of a set of nodes representing variables (e.g. genes) and edges; each edge between two nodes indicates the conditional dependency of the two nodes, given all other nodes. In Gaussian graphical models, the edges between nodes are determined by the non-zero off-diagonal elements in the precision matrix (the inverse of the covariance matrix). Therefore, reconstruction of the graph is equivalent to estimating the precision matrix in the Gaussian graphical model. Friedman et al. (2008) [2] proposed the graphical lasso method to estimate the (inverse) covariance matrices, where they provided an efficient algorithm to directly maximize the L_1-penalized log-likelihood. While the graphical lasso is fast, it only focuses on estimating a single graph. It ignores the structural similarities of multiple graphs when graphical lasso is applied to estimate each graph separately. Recent works aim to recognize possible commonalities among multiple graphs. Peterson et al. (2015) [39] proposed a Bayesian approach to estimate multiple Gaussian graphs by placing a Markov random field prior on the edges and a spike-and-slab prior to control the similarity between graphs. Qiu et al. (2015) [40] proposed a kernel method for joint estimation of multiple Gaussian graphs. Guo et al. [27] proposed to control the sparsity of the off-diagonal elements of the precision matrices and to use the L_1 penalty to control the differences between the off-diagonal elements for each pair of precision matrices. Danaher et al. (2014) [3] proposed the joint graphical lasso algorithm, which uses the L_1 penalty to regularize both the sparsity and the differences between the corresponding off-diagonal elements for each pair of precision matrices. Mohan et al. (2014) [41] extended the joint graphical lasso by taking a node-based approach for estimation of multiple Gaussian graphs. Recently, Zhu et al. (2014) [9] proposed a regularized maximum likelihood method for estimation of multiple precision matrices, In addition to seeking sparseness with a non-convex penalty to regularize the off-diagonal elements in each precision matrix, it also imposes a non-convex

fusion penalty on the differences between each pair of some related precision matrices that can be flexibly specified.

The rest of this paper is organized as follows. In Section 2 we introduce our proposed new methods for estimating component-wise precision matrices in the framework of a Gaussian mixture model. Section 3 presents simulation studies to demonstrate the promising performance of our proposed methods, followed in Section 4 for an application to a glioblastoma gene expression data set. We conclude in Section 5 with a summary of our findings.

2.2 Methods

2.2.1 Gaussian mixture model

We assume that each of n iid p-dimensional observations, $x_1, x_2, \ldots, x_n$, comes from a Gaussian mixture distribution with probability density function

$$f(x_j) = \sum_{i=1}^{g} \pi_i f_i(x_j; \theta_i),$$

where g is the number of components (or populations), π_i is the prior probability for component i with $\sum_{i=1}^{g} \pi_i = 1$, $\theta_i = \{\mu_i, V_i\}$ is the set of the mean and covariance matrix parameters for cluster i, and f_i is a multivariate Normal density (with a component-specific mean μ_i and covariance matrix V_i),

$$f_i(x; \theta_i) = \frac{1}{(2\pi)^{p/2}|V_i|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i)' V_i^{-1}(x-\mu_i)\right).$$

Since each component corresponds to a cluster, we will refer to component and cluster exchangeably. The primary goal here is to estimate the cluster-specific precision matrices $W_i = V_i^{-1}$, though identifying the clusters is often of interest either as a direct or side product.

Given the data, the log-likelihood is

$$\log L(\Theta) = \sum_{j=1}^{n} \log\left(\sum_{i=1}^{g} \pi_i f_i(x_j; \theta_i)\right), \tag{2.1}$$

where $\Theta = \{(\pi_i, \theta_i) : i = 1, 2, ..., g\}$ denotes the set of all unknown parameters. An Expectation-Maximization (EM) algorithm [10] is often used to obtain the maximum

likelihood estimates. For high-dimensional data, it is often beneficial to use the maximum penalized likelihood estimator based on a penalized log-likelihood

$$\log L_P(\Theta) = \log L(\Theta) - p_\lambda(\Theta), \tag{2.2}$$

where $p_\lambda(\Theta)$ is to be specified as a penalty on all or a subset of the parameters. Various penalties have been proposed to achieve better performance in different contexts.

2.2.2 New methods

New method 1: with a convex penalty

A zero entry $W_{i;kl}$, the (k,l)th entry of W_i, indicates conditional independence between the kth and lth variables in cluster i given other variables. Estimating multiple cluster-specific precision matrices can reveal changes of dependency structures across multiple clusters. To facilitate detecting structural changes, a penalty is imposed on the differences between the corresponding entries across multiple precision matrices. We propose using a joint lasso and fused graphical lasso (FGL) penalty of Danaher et al. (2014) [3] on each precision matrices W_i's:

$$p_\lambda(\Theta) = \lambda_1 \sum_{i=1}^{g} \sum_{k \neq l} |W_{i;kl}| + \lambda_2 \sum_{i<i'} \sum_{k,l} |W_{i;kl} - W_{i';kl}|, \tag{2.3}$$

where λ_1 and λ_2 are nonnegative tuning parameters. In addition to achieving sparseness as in graphical lasso, FGL also encourages identical entries across cluster-specific precision matrices. This feature helps to reveal both commonalities and cluster-specific network structures, in addition to improving statistical estimation efficiency through borrowing information across the multiple networks.

Note that Danaher et al. (2014) used the above penalty in the context of Gaussian graphical modeling, knowing which observations are from which Gaussian distribution, differing from our Gaussian mixture modeling. Nevertheless, we will show how to apply their proposed ADMM algorithm ([42]) (as implemented in the R package JGL) in the M-step of an EM algorithm in the current context.

We denote the new method that incorporates the use of the joint lasso and fused graphical lasso (JGL) in our Gaussian mixture modeling as **New-JGL**.

New method 2: with a non-convex penalty

In the context of Gaussian graphical modeling, Zhu et al. (2014) [9] proposed the following non-convex penalty function for W_i,

$$p_\lambda(\Theta) = \lambda_1 \sum_{i=1}^{g} \sum_{k \neq l} J_\tau(|W_{i;kl}|) + \lambda_2 \sum_{i<i'} \sum_{k \neq l} J_\tau(|W_{i;kl} - W_{i';kl}|), \tag{2.4}$$

where λ_1, λ_2 and τ are nonnegative tuning parameters, and $J_\tau(z) = \min(|z|, \tau)$ is the truncated Lasso penalty (TLP) (Shen et al. 2012 [43]). The two penalties serve the corresponding sparseness and fusion roles as in JGL. However, in contrast to FGL in (2.3), only non-diagonal elements, but not diagonal elements, are penalized for their differences in (2.4).

The non-convex TLP reduces the bias induced by the lasso penalty because no more penalty is imposed if $|z| > \tau$ in $J_\tau(z)$. In the current context, the TLP can do better in maintaining the magnitudes of non-zero entries or differences between two unequal entries. The scaled TLP, $J_\tau(z)/\tau$, approximates the L_0-function, $I(z \neq 0)$, as τ tends to 0^+. Like FGL, this method is able to detect possible element-wise heterogeneity across multiple networks, for example in identifying signaling network changes across distinct cancer subtypes.

We propose using the same non-convex penalty (2.4) in our current context of Gaussian mixture modeling, and will demonstrate that the algorithm of Zhu et al. (2014) can be applied in the M-step of an EM algorithm for our purpose. We denote the new method that incorporates the use of structural pursuit (SP) penalty (2.4) in our Gaussian mixture modeling as **New-SP** (New-Structural-Pursuit).

2.2.3 Computing

We develop an EM algorithm to obtain the maximum penalized likelihood estimates (MPLEs). In particular, we will demonstrate how to use an existing Gaussian graphical modeling algorithm in the M-step of the EM algorithm for a penalized Gaussian mixture model.

We introduce z_{ij} as the indicator of whether x_j belongs to component i, so $z_{ij} = 1$ if x_j comes from component i and $z_{ij} = 0$ otherwise. Here z_{ij}'s are treated as missing

data. If z_{ij}'s are observed, the complete data penalized log-likelihood is

$$\log L_{c,P}(\Theta) = \sum_{i=1}^{g}\sum_{j=1}^{n} z_{ij}[\log \pi_i + \log f_i(x_j;\theta_i)] - p_\lambda(\Theta), \tag{2.5}$$

where $p_\lambda(\Theta)$ is a penalty on the parameters; typically only the mean parameters μ_i's and/or covariance matrices V_i's are penalized, which is assumed throughout.

Define the posterior probability of x_j's belonging to component i as $\rho_{ij} = P(z_{ij} = 1|x_j;\Theta)$, then the E-step calculates the following with the current estimate $\Theta^{(r)}$ at iteration r,

$$Q_P(\Theta;\Theta^{(r)}) = E_{\Theta^{(r)}}(\log L_{c,P}|X) = \sum_{i=1}^{g}\sum_{j=1}^{n} \rho_{ij}^{(r)}[\log \pi_i + \log f_i(x_j;\theta_i)] - p_\lambda(\Theta) \tag{2.6}$$

with

$$\hat{\rho}_{ij}^{(r)} = P(z_{ij} = 1|x_j;\Theta^{(r)}) = \frac{\hat{\pi}_i^{(r)} f_i(x_j';\theta_i^{(r)})}{\sum_{i=1}^{g}\hat{\pi}_i^{(r)} f_i(x_j';\theta_i^{(r)})}. \tag{2.7}$$

In the M-step, we find $\hat{\pi}_i^{(r+1)}$, $\hat{\mu}_i^{(r+1)}$ and $\hat{W}_i^{(r+1)}$ that maximize Q_P. Using the Lagrange multiplier η to constrain $\sum_{i=1}^{g}\pi_i = 1$, we omit the terms without π_i's and rewrite Q_P as

$$L(\pi,\eta) = \sum_{i=1}^{g}\sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}\log\pi_i + \eta(\sum_{i=1}^{g}\pi_i - 1). \tag{2.8}$$

Taking the partial derivative of $L(\pi,\eta)$ with respect to π_i and set it to 0, we arrive at the updating formula for $\hat{\pi}_i$

$$\hat{\pi}_i^{(r+1)} = \sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}/n. \tag{2.9}$$

To update μ_i, if there is no penalty on μ_i, we take the derivative of Q_P with respect to μ_i and set it to 0,

$$\frac{\partial Q_P}{\partial \mu_i} = \sum_{j}^{n}\hat{\rho}_{ij}^{(r)}(x_j - \mu_i)'\hat{W}_i = 0, \tag{2.10}$$

obtaining the updating formula for $\hat{\mu}_i$ as

$$\hat{\mu}_i^{(r+1)} = \frac{\sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}x_j}{\sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}}. \tag{2.11}$$

On the other hand, if the Lasso penalty is imposed on μ_i, then its updating formula involves a soft-thresholding on the above quantity (e.g., Pan and Shen 2007 [11]).

Finally, to update V_i or equivalently, $W_i = V_i^{-1}$, we only need to consider the terms related to W_i in Q_P:

$$\begin{aligned} Q_P &= \frac{1}{2}\sum_{i=1}^{g}\sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}\log|W_i| - \frac{1}{2}\sum_{i=1}^{g}\sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}(x_j-\hat{\mu}_i^{(r)})'W_i(x_j-\hat{\mu}_i^{(r)}) - p_\lambda(\Theta) \\ &= \frac{1}{2}\sum_{i=1}^{g}\sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}\left(\log|W_i| - tr(\tilde{S}_i^{(r)}W_i)\right) - p_\lambda(\Theta) \end{aligned} \tag{2.12}$$

with

$$\tilde{S}_i^{(r)} = \frac{\sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}(x_j-\hat{\mu}_i^{(r)})(x_j-\hat{\mu}_i^{(r)})'}{\sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}} \tag{2.13}$$

as a weighted sample covariance matrix.

Typically there is no closed-form solution to update W_i or V_i when one of them is penalized. However, we can take advantage of the existing methods for penalized Gaussian graphical models. Below we point out their connection.

If we know the cluster label for each observation x_j, as in Gaussian graphical modeling, then the penalized log-likelihood for W_i is

$$\frac{1}{2}\sum_{i=1}^{g}\left[n_i\left(\log|W_i| - tr(S_iW_i)\right) - p_\lambda(W_i)\right], \tag{2.14}$$

where n_i is the sample size for cluster i, and S_i is the sample covariance matrix for cluster i. Correspondingly, in the current context of Gaussian mixture modeling, the Q_P function in the EM algorithm with a penalty on W_i is

$$Q_P = \frac{1}{2}\sum_{i=1}^{g}\left[\sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}\left(\log|W_i| - tr(\tilde{S}_i^{(r)}W_i)\right) - p_\lambda(W_i)\right]. \tag{2.15}$$

To maximize Q_P, we use the soft assignment, instead of hard assignment, of each observation x_j into a cluster. Specifically, setting $n_i = \sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}$ and $S_i = \tilde{S}_i^{(r)}$, then maximizing expression (2.15) will be equivalent to maximizing (2.14). Since there are already efficient computational algorithms to maximize the penalized log-likelihood (2.14) in the Gaussian graphical model, we can incorporate one of them into the M-step in our EM algorithm to obtain an update for W_i. Zhou et al. (2009) used this idea in

applying graphical Lasso (Friedman et al. 2008 [2]) in their penalized model-based clustering with unconstrained covariance matrices. We applied the R functions of Danaher et al. (2014) and Zhu et al. (2014) in the M-step for the proposed two new methods respectively.

2.2.4 Review: two existing methods

Different choice of $p_\lambda(W_i)$ will lead to different penalized maximum likelihood estimates of W_i and corresponding algorithms. For comparison, we briefly review two existing penalized mixture modeling methods (Pan and Shen (2007); Zhou et al. (2009) [11, 12]). The method of Pan and Shen (2007) specifies each component in the Gaussian mixture model as a multivariate normal with a common diagonal covariance matrix $V_i = V = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_p^2)$. They proposed an L_1-penalty for the mean parameters,

$$p_\lambda(\Theta) = \lambda_1 \sum_{i=1}^{g} \sum_{k=1}^{p} |\mu_{ik}|, \tag{2.16}$$

where μ_{ik} is the mean of kth variable for component i . Using the L_1 penalty, small estimates of the mean parameters will be shrunken to be exactly zero. If for a given variable k, $\mu_{ik} = 0$ for all components i, then this variable will have no effect on clustering. Hence this penalty is used for variable selection, but not for inferring the cluster-specific networks.

Zhou et al. (2009) [12] relaxed the diagonal covariance matrix assumption and adopted unconstrained covariance/precision matrices. To regularize the parameters in the precision matrices, they proposed a penalty function of the form

$$p_\lambda(\Theta) = \lambda_1 \sum_{i=1}^{g} \sum_{k=1}^{p} |\mu_{ik}| + \lambda_2 \sum_{k \neq l} |W_{kl}|, \tag{2.17}$$

where $W = V^{-1}$ is the common precision matrix (or inverse covariance matrix). The first term in the above penalty function aims at variable selection as in Pan and Shen (2007), while the second term uses the L_1-penalty to promote the sparseness of the precision matrix. For penalized covariance matrix estimation, they used the graphical lasso algorithm of Friedman et al. (2008) and maximized the following objective function

$$\log |W| - tr(\tilde{S}^{(r)} W) - \lambda \sum_{k \neq l} |W_{kl}|, \tag{2.18}$$

where $\lambda = 2\lambda_2/n$ and

$$\tilde{S}^{(r)} = \frac{\sum_{i=1}^{g}\sum_{j=1}^{n}\hat{\rho}_{ij}^{(r)}(x_j - \hat{\mu}_i^{(r)})(x_j - \hat{\mu}_i^{(r)})'}{n}$$

is a weighted sample covariance matrix based on the soft assignments of all the samples as for $\tilde{S}_i^{(r)}$.

Zhou et al. (2009) also considered the case where each component i in the mixture model has an unconstrained covariance matrix V_i. Then they proposed the following penalty function to regularize the means and cluster-specific covariance matrices,

$$p_\lambda(\Theta) = \lambda_1 \sum_{i=1}^{g}\sum_{k=1}^{p}|\mu_{ik}| + \lambda_2 \sum_{i=1}^{g}\sum_{j,l}^{p}|W_{i;j,l}|, \tag{2.19}$$

where $W_{i;j,l}$ is the (j,l)th entry of W_i. They again used the graphical lasso algorithm to obtain the estimate of the cluster-specific precision matrix $W_i = V_i^{-1}$ in the M-step of the EM algorithm.

2.2.5 Implementation

By default our EM algorithm starts with some initial values given by the K-means method, though other (random or fixed) and/or multiple starting values can be equally applied.

We first use the L_1-penalty, then try τ at each of the quantiles of the L_1-penalized estimates of $|W_{i;kl}|$ and $|W_{i;kl} - W_{i';kl}|$. By default the tuning parameters λ_1 and λ_2 are chosen from $\lambda_1 \in \{\log(p) \times (1.5, 1, 0.8, 0.3, 0.1, 0.05, 0.01, 0.001)\}$ and $\lambda_2 \in \{\log(p) \times (10^8, 1000, 500, 100, 50, 10, 5, 1, 0.8, 0.5, 0.3, 0.1, 0.01, 0.001)\}$. A grid search is used to find a combination of the penalty parameter values (λ_1, λ_2, τ) and a cluster number g that lead to the highest predictive log-likelihood as calculated by 5-fold cross-validation.

Our methods are implemented in an R package called pGMM that will be freely downloadable on CRAN.

2.3 Simulations

Due to the unknown truth for real data, it is difficult to draw definitive conclusions on the relative performance of various methods. As an alternative, we conducted simulations

to evaluate and compare the performance of the methods in both clustering (i.e. the assignments of the samples to clusters) and precision matrix estimation.

To mimic real data, we used the fitted models to the glioblastoma gene expression data by Zhou et al. (2009) and our proposed new methods as the true model to generate simulated data; in this way, we avoided possible biases in using only one true model to generate simulated data that might favor one of the methods. In each case, there were 4 clusters with $n = 173$ or $n = 346$ observations with $p = 20$. We then applied the usual non-penalized model-based clustering as implemented in R package mclust (Fraley and Raftery 2006 [29]), the methods of Pan and Shen (2007) [11] and Zhou et al. (2009) [12], and our proposed two new methods. To measure the accuracy of parameter estimation for precision matrices, we used the average entropy loss (EL) and average quadratic loss (QL),

$$EL = \frac{1}{g}\sum_{i=1}^{g}\left(tr(V_i\hat{W}_i) - \log\det(V_i\hat{W}_i)\right)$$
$$QL = \frac{1}{g}\sum_{i=1}^{g} tr\left((V_i\hat{W}_i - I)^2\right).$$

To measure the accuracy of estimating zero or non-zero entries and grouping structures in precision matrices, following Zhu et al. (2014) [9], we used the average number of false positives for sparseness pursuit (FPV), average number of false negatives for sparseness pursuit (FNV), average number of false positives for grouping (FPG), and average number of false negatives for grouping (FNG):

$$FPV = \frac{1}{g}\sum_{i=1}^{g}\frac{\sum_{1\le j\le j'\le K}\mathbb{I}(W_{i;jj'}=0,\hat{W}_{i;jj'}\neq 0)}{\sum_{1\le j\le j'\le K}\mathbb{I}(W_{i;jj'}=0)}\times\Big(1-\mathbb{I}(W_{i,\text{off}}\neq 0)\Big)$$

$$FNV = \frac{1}{g}\sum_{i=1}^{g}\frac{\sum_{1\le j\le j'\le K}\mathbb{I}(W_{i;jj'}\neq 0,\hat{W}_{i;jj'}=0)}{\sum_{1\le j\le j'\le K}\mathbb{I}(W_{i;jj'}\neq 0)}\mathbb{I}(W_{i,\text{off}}\neq 0)$$

$$FPG = \frac{1}{C(g,2)}\sum_{i<i'}\frac{\sum_{1\le j\le j'\le K}\mathbb{I}(W_{i;jj'}=W_{i';jj'},\hat{W}_{i;jj'}\neq\hat{W}_{i';jj'})}{\sum_{1\le j\le j'\le K}\mathbb{I}(W_{i;jj'}=W_{i';jj'})}$$
$$\times\Big(1-\mathbb{I}(W_{i,\text{off}}\neq W_{i',\text{off}})\Big)$$

$$FNG = \frac{1}{C(g,2)}\sum_{i<i'}\frac{\sum_{1\le j\le j'\le K}\mathbb{I}(W_{i;jj'}\neq W_{i';jj'},\hat{W}_{i;jj'}=\hat{W}_{i';jj'})}{\sum_{1\le j\le j'\le K}\mathbb{I}(W_{i;jj'}\neq W_{i';jj'})}$$
$$\times\mathbb{I}(W_{i,\text{off}}\neq W_{i',\text{off}}),$$

where $C(g,2)$ is the combinatorial number of choosing 2 from g.

Table 2.1 shows the results for $n = 173$ based on 50 simulations for each set-up. With the true model as the fitted model by the method of Zhou et al. (2009), the method of Zhou et al. (2009) itself gave the highest Rand index, suggesting the best accuracy for clustering. However, It did not give the lowest average entropy loss (EL) and quadratic loss (QL) for precision matrix estimation, though the differences were not large. Recall that the true model here was based on four largely differing cluster-specific precision matrices, which might not favor fusing the cluster-specific precision matrices. Impressively our method New-SP gave the second highest Rand index that was quite close to that of Zhou et al. (2009), and more importantly, New-SP gave the most or second most accurate estimates of the cluster-specific precision matrices with the lowest average EL and second lowest QL. In addition, it also gave low false positive rates of sparseness and grouping, but high false negative rates. It is noted that New-JGL also performed well.

On the other hand, if the true model was the fitted one by New-SP, then New-SP was the clear winner for both clustering and precision matrix estimation, followed by New-JGL. This was the case when the cluster-specific precision matrices differed but sharing some commonalities. Finally, if the true model was the fitted one from New-JGL, the winners were New-JGL and New-SP, followed by mclust and the method of

Truth	Method	RI	aRI	EL	QL	FPV	FNV	FPG	FNG
Zhou09	Zhou09	**0.714**	**0.309**	30.495	61.121	0.749	0.204	0.875	0.110
		(0.036)	(0.087)	(1.330)	(21.484)	(0.026)	(0.021)	(0.081)	(0.084)
	Pan07	0.648	0.164	29.569	66.893	0.000	1.000	0.000	1.000
		(0.033)	(0.054)	(0.208)	(4.349)	(0.000)	(0.000)	(0.000)	(0.000)
	Mclust	0.632	0.159	30.054	70.533	0.000	1.000	0.000	1.000
		(0.027)	(0.044)	(0.620)	(11.465)	(0.000)	(0.000)	(0.000)	(0.000)
	New-JGL	0.660	0.189	29.049	**59.005**	0.127	0.817	0.000	1.000
		(0.026)	(0.054)	(0.673)	(11.801)	(0.174)	(0.182)	(0.000)	(0.000)
	New-SP	0.689	0.260	**28.726**	59.695	0.020	0.952	0.022	0.957
		(0.034)	(0.071)	(0.881)	(11.112)	(0.042)	(0.086)	(0.043)	(0.075)
New-SP	Zhou09	0.632	0.024	31.609	61.775	0.649	0.256	0.881	0.000
		(0.024)	(0.010)	(1.138)	(17.245)	(0.047)	(0.023)	(0.058)	(0.000)
	Pan07	0.804	0.501	33.785	49.081	0.000	1.000	0.000	0.000
		(0.043)	(0.107)	(0.612)	(8.787)	(0.000)	(0.000)	(0.000)	(0.000)
	Mclust	0.862	0.647	26.179	72.099	1.000	0.000	0.319	0.000
		(0.047)	(0.114)	(5.333)	(209.858)	(0.000)	(0.000)	(0.470)	(0.000)
	New-JGL	0.856	0.629	23.853	11.255	0.376	0.292	0.000	0.000
		(0.038)	(0.098)	(0.362)	(2.275)	(0.124)	(0.069)	(0.000)	(0.000)
	New-SP	**0.917**	**0.785**	**23.265**	**10.242**	0.130	0.558	0.000	0.000
		(0.053)	(0.136)	(0.977)	(4.657)	(0.072)	(0.106)	(0.000)	(0.000)
New-JGL	Zhou09	0.664	0.063	30.174	46.403	0.692	0.245	0.911	0.090
		(0.027)	(0.017)	(0.769)	(8.676)	(0.029)	(0.018)	(0.037)	(0.040)
	Pan07	0.886	0.717	30.283	31.125	0.000	1.000	0.000	1.000
		(0.036)	(0.087)	(0.624)	(5.356)	(0.000)	(0.000)	(0.000)	(0.000)
	Mclust	0.897	0.744	25.875	24.760	0.020	0.980	0.000	1.000
		(0.030)	(0.072)	(0.392)	(3.434)	(0.141)	(0.141)	(0.000)	(0.000)
	New-JGL	0.926	0.815	**22.533**	**7.615**	0.361	0.341	0.000	1.000
		(0.043	(0.109)	(0.258)	(1.237)	(0.047)	(0.048)	(0.000)	(0.000)
	New-SP	**0.930**	**0.823**	23.493	12.694	0.056	0.759	0.000	1.000
		(0.042)	(0.107)	(0.786)	(5.237)	(0.049)	(0.159)	(0.000)	(0.000)

Table 2.1: Simulation results with $n = 173$ and the true model being that estimated byone of the three methods based on the glioblastoma dataset. The means (standard deviations) of the Rand Index (RI), adjusted Rand Index (aRI), average entropy loss (EL) average quadratic loss (QL), average false positive for sparseness pursuit (FPV), average false negative for sparseness pursuit (FNV), average false positive for grouping (FPG) and average false negative for grouping (FNG) are shown for 50 simulations.

Pan and Shen (2007) (where a common diagonal precision matrix was assumed).

We also investigated the sensitivity of the EM algorithm to its starting values. For the set-up with the true model as the one fitted by New-JGL, instead of using the K-means output as the starting value for New-JGL and New-SP, we used some randomly generated numbers as the starting value. The resulting Rand index values for New-JGL and New-SP decreased from 0.926 and 0.930 to 0.635 and 0.757 respectively, confirming the importance of using good starting values for the EM algorithm. However, the estimation errors for the precision matrices were less influenced: for example, the mean EL for New-JGL and New-SP increased from 22.533 and 23.493 to only 23.964 and 25.743, respectively, still lower than those of the other methods.

Next we doubled the sample size in simulations. With the increased sample size, the proposed method New-SP became the clear overall winner, followed by New-JGL

(Table 2.2). Although mclust performed well in the last two set-ups (with the true model being that fitted by New-SP or New-JGL), it did not work well in the first set-up. Again it is noted that the two new methods largely outperformed the method of Zhou et al. (2009) [12] for estimating the cluster-specific precision matrices, perhaps due to the former two's use of the fusion penalties for information borrowing across multiple cluster-specific precision matrices.

2.4 Example

2.4.1 Glioblastoma Gene Expression Data

Verhaak et al. (2010) [1] studied a gene expression data set of glioblastoma tumor and normal samples. They used a consensus hierarchical clustering method to identify four disease subtypes. It is noted that, due to the limitation of the clustering method, the conditional dependencies between genes in each cluster were ignored and thus not revealed. This leaves room for our new and other methods to explore possible dependency relationships among the genes. Furthermore, the identified four clusters, albeit biologically reasonable, are in no way to be perfect, which bears importance when one uses their sample assignments as a reference to compare various methods.

To be practically focused, we restricted our analysis to the gene expression data from the 173 core samples as used by Verhaak et al.(2010) [1], and we selected only 20 genes that are related to cell signaling pathways. Some of these genes were demonstrated to be altered in Figure 4B in Brennan et al. (2013) [44] and Figure 5 in Mclendon et al. (2008) [45]. Specifically, genes EGFR, PDGFRA, FGFR3 are members of the RTK signaling pathway. RASGRP3 and RRAS are downstream targets of the RTK signaling pathway. PIK3C2B, PIK3R1, PIK3R3, PIK3IP1 and AKTIP are components of the PI3K/AKT signaling pathway. NFIB is the downstream target of RTK and PI3K/AKT signaling pathways. CDKN3, CDK4, CDKN1A, CDKN2C, CCND2 are involved in RB signaling pathways and they play important roles in cell cycle regulation. CASP1 and CASP4 are important genes in cell apoptosis.

2.4.2 Estimated networks

We applied our method New-SP to the glioblastoma gene expression data set. Trying with $g = 1, 2, 3, 4, 5$ clusters, it reached four clusters/subtypes. Each cluster showed a distinct conditional dependency structure among the genes, though their overall structures were similar (Figure 2.1).

This suggests distinct cell signalling network changes across the disease subtypes. A closer examination of the estimated precision matrices reveals that the conditional dependencies among the receptor kinases and the downstream target genes were altered. The PI3K/Akt signaling pathway plays an important role in cell survival and proliferation in glioblastoma ([46, 47]). One of the estimated networks shows that the AKTIP gene was conditionally correlated with CDKN2C, a gene encoding a cyclin-dependent kinase inhibitor that regulates cell growth. However, this link was lost in all other three estimated networks. Similarly, PIK3IP and AKTIP were not conditionally correlated with CDKN1A, CDKN2C and CDKN3 in one or more estimated networks, while the network in bottom left of Figure 2.1 preserved most of the connections. The PI3K/Akt signaling pathway is reported to be upstream of CCND2, a gene encoding the cell cycle regulating protein Cyclin D2 ([48]). Only one out of four subtypes demonstrated a conditional dependence between AKTIP and CCND2. The changes between these links collectively suggested dysregulation of cell growth by the PI3K/Akt signaling pathway in some subtypes of glioblastoma.

Gene IDH1 is known to have a higher mutation frequency in some glioblastoma subtypes, and here it exhibited cluster-specific associations with FGFR3, which was also reported to have mutations in glioblastoma subtypes classified by Verhaak et al. (2010) [1]. We found that gene IDH1's expression was positively correlated with that of FGFR3 in only one cluster, suggesting possibly altered co-expressions in other clusters. IDH1 mutation is reported to cause widespread changes in histone and DNA methylation and potentially promoting tumorigenesis ([49, 50]). CCND2 was found to be amplified in IDH1 mutant medulloblastoma subtypes ([51]). Therefore, the abnormal IDH1 gene level and its disconnection with CCND2 observed in the estimated network pointed to possible roles of IDH1 in oncogenesis in certain subtypes of glioblastoma.

For comparison, we applied Zhou et al.'s method to the glioblastoma gene expression data set with cluster-specific covariance matrices. Among $g = 1, 2, 3, 4, 5$ clusters, it

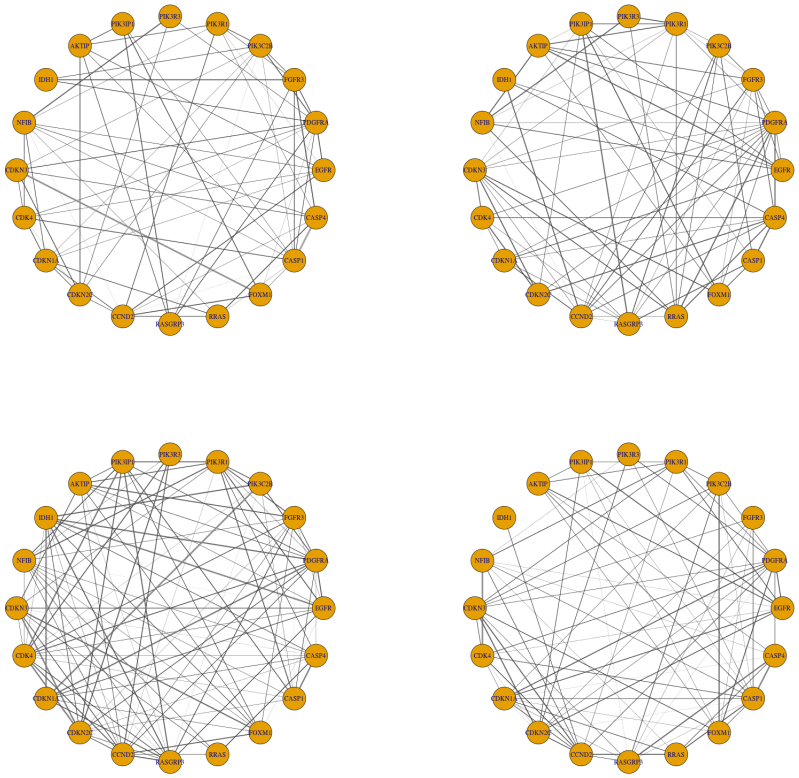

Figure 2.1: Estimated cluster-specific networks based on 173 core samples using the new method New-SP.

selected four clusters. The estimated cluster-specific precision matrices demonstrated cluster-specific dependencies among the genes (Figure 2.2). The estimated networks using Zhou et al.'s method confirmed that the conditional correlation between IDH1 and CCND2 was lost in one network estimated by the New-SP method. The conditional correlation between AKTIP and CCND2 was present in three subtypes, though the correlation was weak in one subtype. Compared to the networks estimated by the method of New-SP, the dependency changes across the clusters estimated by Zhou et al.'s method were much more dramatic, reflecting possibly large variations of the estimates without borrowing information across clusters.

The New-JGL method also yielded four clusters (Figure 2.3). Like the networks

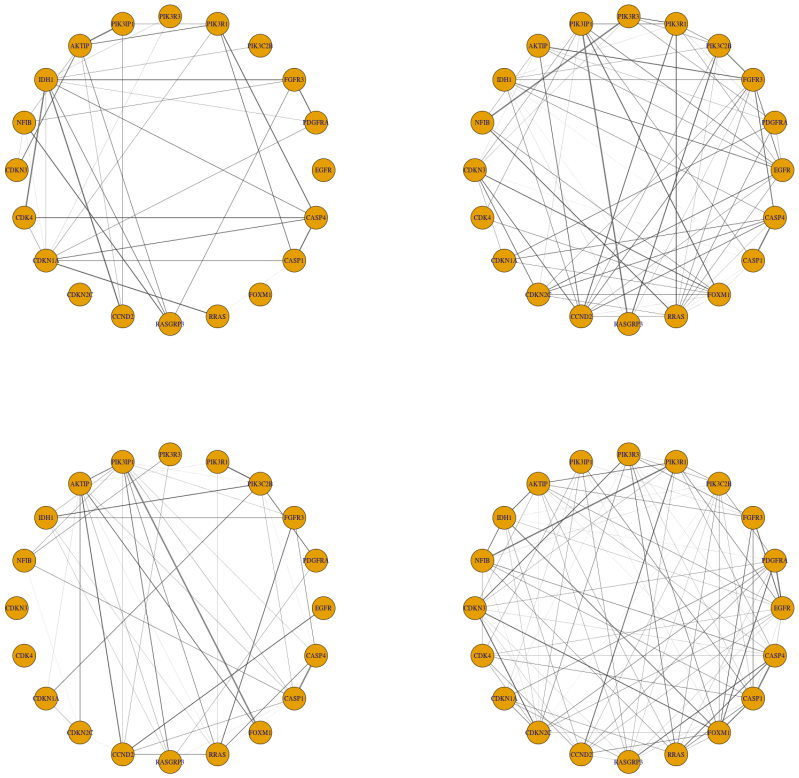

Figure 2.2: Estimated cluster-specific networks based on 173 core samples using the method of Zhou et al. (2009).

estimated by the New-SP method, the networks estimated by the New-JGL method shared some structural similarity. The AKTIP and CCND2 correlation was found in two out of four subtypes, although the correlation in one subtype was weak. This agreed with the correlation in the networks estimated by the New-SP method. Unlike in Figure 2.1, the conditional correlation between IDH2 and CCND2 was present in all four subtypes, though the magnitude of correlation was small.

The non-penalized mclust yielded four clusters with a common covariance matrix, suggesting that the differences among the four cluster-specific covariance matrices were possibly subtle. This was also reflected from the overall similarity across the four cluster-specific estimates of the two new methods.

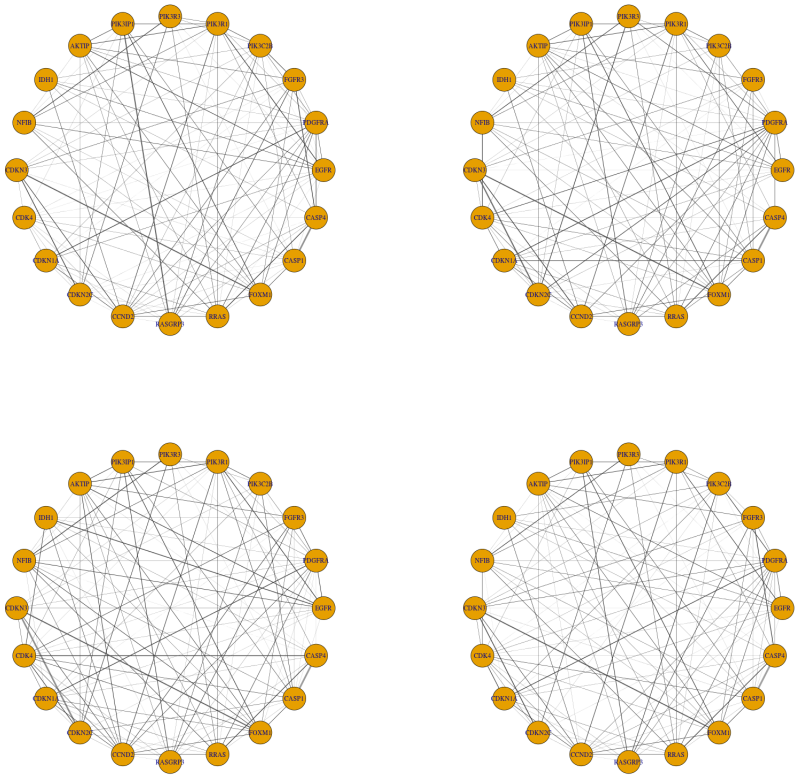

Figure 2.3: Estimated cluster-specific networks based on 173 core samples using the method of New-JGL.

2.4.3 Sample cluster assignments

Using the cluster assignments in Verhaak et al. (2010) [1] as the reference, the agreement between our New-SP method and the reference as measured by the Rand index was 0.747 and by the adjusted Rand index was 0.354 (Table 2.3). Its performance was compared with several other methods. First, the method of Pan and Shen (2007) [11] based on a common diagonal covariance matrix yielded 5 clusters with a Rand index of 0.749 and the adjusted Rand index of 0.358. For the purpose of comparison, we also examined the clustering results of the method by forcing 4 clusters, which led to a Rand index of 0.780 and the adjusted Rand index of 0.439 (Table 2.3). Although the method of Pan and Shen yielded a slightly higher Rand index, it was possibly due to the bias of

the reference clustering method (that ignored varying within-cluster dependencies that would in turn favor the results of Pan and Shen (2007)). More importantly, a common diagonal covariance matrix assumed and estimated by the method cannot be used to examine possibly varying within-cluster dependency structures. Finally, the two new methods seemed to perform better than the two other methods.

2.4.4 Model assessment

To check the goodness-of-fit of a final model, we propose using the parametric bootstrap, which was used by McLachlan and others to select the number of components in a Gaussian mixture model ([52, 53]). For example, for our real data, the New-SP method selected a final model with four components, each with a component-specific precision matrix, which is called an alternative model here; it may be of interest to compare this alternative model with a null (or reduced) model with four components but a common precision matrix, which could be achieved by forcing a large λ_2 value (while other tuning parameters were selected as before). We generated 50 bootstrap samples from the fitted null and alternative models respectively, then fitted the two models respectively to the bootstrap samples; finally, we compared their corresponding CV log-likelihood values, as shown in Figure 2.4. For the bootstrap samples, in both cases fitting the alternative model seemed to yield a higher mean value of the CV log-likelihood; however, the difference between the two fitted models was larger when the bootstrap samples were generated from the alternative model, as expected. Since the CV log-likelihood value difference between the two fitted models based on the original data was larger than that from the bootstrap samples generated from the alternative model, there was some evidence to support the use of the alternative model. Nevertheless, perhaps due to the relatively small sample sizes and shrinkage effects of the four component-specific precision matrices towards each other (as imposed by the fusion penalty even in the alternative model), the difference between the two models was not overwhelming, and cautions must be taken in not over-interpreting their differences.

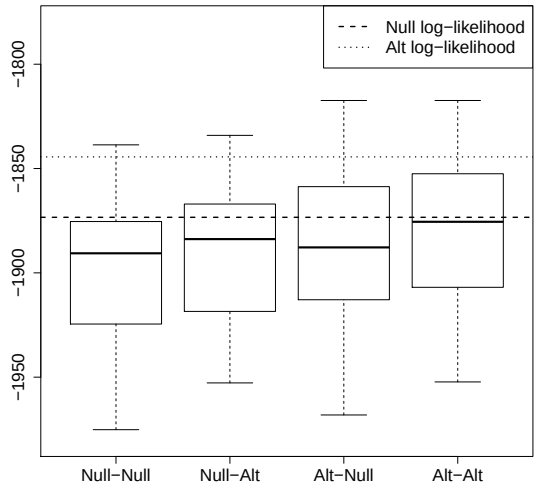

Figure 2.4: Distributions of the CV log-likelihood values of various fitted models based on bootstrap samples. Null-Null, bootstrap samples were generated from the null model, to which the null model was fitted; Null-Alt, bootstrap samples were generated from the null model, to which the alternative model was fitted; Alt-Null, bootstrap samples were generated from the alternative model, to which the null model was fitted; Alt-Alt, bootstrap samples were generated from the alternative model, to which the alternative model was fitted. The two horizontal lines are the CV log-likelihood values for the two fitted models to the original data.

2.5 Discussion

We have presented a new approach to estimation of multiple networks in the context of a penalized Gaussian mixture model. The primary goal is for estimating and comparing cluster-specific network changes, though automatic cluster discovery is often of interest too. For the primary goal, it is necessary to encourage the equalities of the entries across the cluster-specific precision matrices while maintaining their differences if any, which is best accomplished by fusion with a non-convex penalty such as TLP as adopted in our proposed method New-SP ([43], [9]). Note that standard and existing penalized model-based clustering methods are not suitable for our primary goal: due to the lack of fusion penalties, the existing methods cannot highlight few major differneces across multiple precision matrix estimates, in addition to their loss of estimation efficiency

without information borrowing. Both our proposed methods pursue both sparseness and fusion for multiple precision matrices in the framework of Gaussian mixture modeling. Our approach takes advantage of the existing methods using convex or non-convex penalties to regularize the parameters in the unconstrained precision matrices based on Gaussian graphical models, which assumes that it is known that which samples are from which Gaussian distributions, differing from our current context with unknown sample heterogeneity.

We applied the methods to a real data set containing gene expression profiles of glioblastoma patients. Using the New-SP method, the samples were partitioned into four disease subtypes, as reported in Verhaak et al. (2010) but based on only differential gene expression. Importantly, our method reconstructed disease subtype-specific gene networks, suggesting candidates for possibly subtype-specific gene dysregulations that can be followed up in further biological experiments. Since the truth is unknown for the real data, we recoursed to realistic simulations mimicking the real data to evaluate the methods; it was demonstrated that our method New-SP based on the non-convex TLP gave the best overall performance in both clustering (i.e. subtype discovery) and network estimation when the sample size was at least moderately large, followed by the other proposed method New-JGL based on the convex (fused) Lasso penalty. The better performance of New-SP over New-JGL is likely due to the non-convex TLP adopted in the former, as demonstrated in Shen et al. (2012) [43] for regression and single precision matrix estimation and Zhu et al. (2014) [9] for estimating multiple precision matrices in Gaussian graphical models. On the other hand, New-JGL is simpler and faster than New-SP, and thus can be used for larger problems and/or to provide a quick preliminary solution; in particular, we advocate using the results of New-JGL (or any other method with a convex penalty) as a good starting value for New-SP, thus the latter can be regarded as a refinement of the former. We also note that partition rules discussed in Zhu et al. (2014) can be used to speed up the new methods for high-dimensional data.

We emphasize that the existing methods for estimation of multiple networks, including the two used here (Danaher et al. 2014; Zhu et al. 2014 [3, 9]), are based on Gaussian graphical models without sample heterogeneity; that is, each sample is assumed to be known from a given Gaussian distribution. In our target applications and other settings, this sample homogeneity assumption may not hold. For example, in

clinical genomic studies, due to disease heterogeneity, the assumption that all the gene expression profiles of cancer patients come from the same Gaussian distribution is not practical. To discover unknown disease subtypes, clustering or unsupervised learning becomes useful, which will facilitate personalized medicine. To our knowledge, existing clustering methods of gene expression have focused on detecting differential mean expression levels across clusters or disease subtypes, as demonstrated in Verhaak et al. (2010) [1]. However, in addition to differential gene expression, there are possibly differential gene regulations or dysregulations across disease subtypes. If disease subtypes are known, then differential gene regulations can be treated as estimating multiple precision matrices in Gaussian graphical models, as handled by many existing methods; otherwise, as discussed here, both disease subtypes and possibly differential precision matrices must be inferred simultaneously based on a Gaussian mixture model.

Our methods are implemented in an R package pGMM that will be available on CRAN.

Acknowledgment

The authors are grateful to the Editor and reviewers for constructive comments. This research was supported by NIH grants R01GM081535, R01GM113250, R01HL105397 and R01HL116720, and by the Minnesota Supercomputing Institute.

Truth	Method	RI	aRI	EL	QL	FPV	FNV	FPG	FNG
Zhou09	Zhou09	0.775	0.455	26.072	24.435	0.825	0.115	0.936	0.049
		(0.040)	(0.097)	(1.185)	(12.397)	(0.021)	(0.022)	(0.065)	(0.062)
	Pan07	0.608	0.123	29.191	56.655	0.000	1.000	0.000	1.000
		(0.036)	(0.049)	(0.140)	(2.808)	(0.000)	(0.000)	(0.000)	(0.000)
	Mclust	0.675	0.239	29.647	72.100	0.000	1.000	0.000	1.000
		(0.040)	(0.079)	(1.543)	(28.488)	(0.000)	(0.000)	(0.000)	(0.000)
	New-JGL	0.661	0.200	27.316	34.188	0.583	0.329	0.007	0.992
		(0.022)	(0.043)	(0.356)	(4.787)	(0.028)	(0.033)	(0.050)	(0.057)
	New-SP	**0.842**	**0.624**	**24.502**	**17.738**	0.128	0.605	0.206	0.541
		(0.052)	(0.129)	(1.163)	(7.946)	(0.032)	(0.067)	(0.054)	(0.074)
New-SP	Zhou09	0.893	0.726	29.645	42.974	0.743	0.156	0.956	0.000
		(0.032)	(0.083)	(0.587)	(7.665)	(0.041)	(0.016)	(0.033)	(0.000)
	Pan07	0.830	0.570	34.003	49.082	0.000	1.000	0.000	0.000
		(0.025)	(0.061)	(0.354)	(5.454)	(0.000)	(0.000)	(0.000)	(0.000)
	Mclust	0.922	0.802	24.722	23.761	1.000	0.000	0.950	0.000
		(0.037)	(0.094)	(1.704)	(18.492)	(0.000)	(0.000)	(0.198)	(0.000)
	New-JGL	0.883	0.701	22.864	7.019	0.426	0.185	0.000	0.000
		(0.037)	(0.095)	(0.234)	(1.33)8	(0.103)	(0.040)	(0.000)	(0.000)
	New-SP	**0.962**	**0.902**	**21.035**	**2.534**	0.544	0.106	0.000	0.000
		(0.018)	(0.048)	(0.318)	(0.827)	(0.230)	(0.116)	(0.001)	(0.000)
New-JGL	Zhou09	0.931	0.827	28.631	35.451	0.788	0.153	0.963	0.036
		(0.025)	(0.063)	(0.618)	(6.236)	(0.022)	(0.013)	(0.007)	(0.011)
	Pan07	0.898	0.747	30.255	30.215	0.000	1.000	0.000	1.000
		(0.023)	(0.056)	(0.507)	(3.997)	(0.000)	(0.000)	(0.000)	(0.000)
	Mclust	0.941	0.853	22.358	7.122	1.000	0.000	0.275	0.725
		(0.021)	(0.052)	(0.610)	(2.899)	(0.002)	(0.000)	(0.446)	(0.446)
	New-JGL	0.938	0.845	21.844	4.899	0.458	0.216	0.104	0.869
		(0.024)	(0.060)	(0.130)	(0.650)	(0.069)	(0.055)	(0.241)	(0.304)
	New-SP	**0.961**	**0.903**	**21.637**	**4.359**	0.109	0.540	0.000	0.998
		(0.011)	(0.028)	(0.171)	(0.698)	(0.116)	0.105)	(0.001)	(0.006)

Table 2.2: Simulation results with $n = 346$ and the true model being that estimated byone of the three methods based on the glioblastoma dataset. The means (standard deviations) of the Rand Index (RI), adjusted Rand Index (aRI), average entropy loss (EL) average quadratic loss (QL), average false positive for sparseness pursuit (FPV), average false negative for sparseness pursuit (FNV), average false positive for grouping (FPG) and average false negative for grouping (FNG) are shown for 50 simulations.

		New-SP	mclust	Pan07	Zhou09	New-JGL
$p = 20$	RI	0.747	0.688	0.780	0.713	0.746
	aRI	0.354	0.222	0.439	0.305	0.355

Table 2.3: Rand Index (RI) and adjusted Rand Index (aRI) for the glioblastoma gene expression data with 20 genes by various methods. The class assignments given in [1] are used as the reference.

Chapter 3

ADAPTIVE TESTING OF SNP-BRAIN FUNCTIONAL CONNECTIVITY ASSOCIATION VIA A MODULAR NETWORK ANALYSIS

3.1 Introduction

Resting-state functional magnetic resonance imaging (rs-fMRI) is gaining popularity in studies of brain functional connectivity with applications to detection of subtle network reorganizations in Alzheimer's disease [54]. Disruption of connectivity in the brain functional network is related to many pathological conditions in the brain, such as Alzheimer's disease [55], schizophrenia [15], or autism [17]. This necessitates the development of methods for modelling the brain functional network its statistical inference.

A network is comprised of nodes and edges connecting the nodes. Based on functional MRI data, a popular choice of nodes are brain regions of interest (ROIs) while the

edges are connectivities reflecting statistical dependencies between ROIs. An important network model, the scale-free network[56], assumes that most nodes in a network are sparsely connected with the exception of a few "hub" nodes that are densely connected with other nodes. In the scale-free network model, new connections are more likely to occur for those hub nodes with already-high connectivity. There has been empirical evidence supporting this model for brain functional networks [57], though it is still debatable. In addition, the scale-free network model also admits a modular topological structure, which can be extracted for more efficient analyses for human brains.

Methods for drawing statistical inference to distinguish brain connectivity for different groups of subjects are still under development. The first question encountered is how to define brain functional connectivity. Ref. [58] discussed the choice between Pearson's marginal correlation coefficient and partial correlation coefficient as a network connectivity measure, though other measures are possible and it is yet unclear which one is best. To reduce dimensionality and to reach sparseness, graphical lasso is often used for estimating networks for different groups. Since an estimated network with the imposed sparsity penalty may not demonstrate modular structures, a better approach is to directly discover the modules in a network. A general framework for estimating scale-free networks and detecting modules is proposed in Ref. [20] for gene network analysis, which has gained tremendous popularity in genomics [59]. It starts by defining a similarity measure between two nodes in a network, called adjacency, using the marginal correlation coefficient. Soft-thresholding is then applied, leading to a weighted network. The soft-thresholded adjacency is further transformed to a topological overlap matrix (TOM) element, which is converted to a dissimilarity measure for hierarchical clustering, grouping closely connected nodes together as modules in the network. The above framework not only provides multiple network connectivity measures, but also carries out modular structure identification. The connectivity measures and identified modules in the brain functional network may help statistical inference and offer biological insights [59].

In this paper, for the first time, we adapt the use of WGCNA for gene expression data to rs-fMRI data, constructing weighted brain functional networks and identifying their subnetworks or modules using the Alzheimer's Disease Neuroimaging Initiative (ADNI) data. We explored using the adjacency matrix element and TOM element, in

addition to the marginal correlation or covariance, to characterize connectivity in brain functional networks. Taking advantages of detected network modules, we conduct association analysis of genetic variants with not only the whole brain functional network, but also its various subcomponents, including its modules, which aims to not only improve statistical power, but also offer better biological interpretation. We propose applying a new adaptive association test based on a proportional odds model (POM) accounting for the ordinal nature of the SNP genotype. We found evidence of associations between several network modules and the APOE4 variant, which is by far the most significant genetic risk factor for Alzheimer's disease.

This paper is organized as follows. We first review the method of WGCNA, including its module identification, then introduce the adaptive test based on a POM. We demonstrate the application of our methods to the ADNI data before summarizing our findings and future research directions in the discussion section.

3.2 Methods

3.2.1 Module detection via weighted gene co-expression network analysis

In this section, we briefly review the work in Ref. [20] on the weighted gene-coexpression network analysis (WGCNA) framework for network construction and module identification.

Adjacency matrix

The first step of the WGCNA framework is to define a similarity measure between gene expression profiles; in the current context, we use the BOLD signals in each of multiple ROIs from one or more subjects to calculate a similarity between any two ROIs. The similarity measure is required to take values between 0 and 1. A typical choice of this similarity measure is the absolute value of the Pearson correlation coefficient $s_{uv} = |cor(u, v)|$, for nodes u and v. Another choice, which preserves the sign of correlation, is defined as $s_{uv} = [1 + cor(u, v)]/2$. We refer the first one as unsigned similarity measure, and the second one as the signed similarity measure. From our experience of

applications to the ADNI data, the identified modules have negligible differences using either unsigned or signed similarity measure. We used the unsigned similarity measure throughout this paper.

Once the similarity measure is computed, the next step is to transform the similarity matrix $S = [s_{uv}]$ into an adjacency matrix using an adjacency function. Hard thresholding is often used to yield a binary or unweighted network with a 0/1 adjacency indicating no-connection/connection and thus possible loss of information, though a more efficient multi-scale approach with multiple thresholds yielding a set of binary networks has been proposed [60]. Soft thresholding is a simple and popular alternative with more flexibilities. One choice is the power adjacency function

$$a_{uv} = power(s_{uv}, \delta) \equiv |s_{uv}|^{\delta} \tag{3.1}$$

with parameter δ, which is chosen as the smallest integer such that the scale-free network model fitting is above a certain threshold.

Topological overlap matrix

Instead of using only the adjacency matrix, Ref. [61] advocated a topological overlap matrix $\Omega = [\omega_{uv}]$ with its element as a potentially more useful measure that reflects the relative interconnectedness of two nodes u and v after accounting for their shared neighbors. The topological overlap matrix element is defined as

$$\omega_{uv} = \frac{l_{uv} + a_{uv}}{\min\{k_u, k_v\} + 1 - a_{uv}} \tag{3.2}$$

with $k_u = \sum_v a_{uv}$ and $l_{uv} = \sum_q a_{uq} a_{qv}$. For a binary network with $a_{uv} = 0$ or 1, k_u is the connectivity of node u representing the number of its direct neighbors, while l_{uv} equals the number of nodes that connect both nodes u and v; $\omega_{uv} = 0$ if the nodes u and v are not connected and they are not connected to the same neighbors; in contrast, $\omega_{uv} = 1$ if the nodes u and v are connected and the neighbors of the node with fewer edges are also connected to the one with more edges. For any network, $0 \leq a_{uv} \leq 1$ implies $0 \leq \omega_{uv} \leq 1$.

Module identification

To identify modules in a network, we need to have a dissimilarity or distance measure. An intuitive way is to convert a similarity measure. Based on the topological overlap matrix element ω_{uv}, we can simply define the dissimilarity measure as $d^{\omega}_{uv} = 1 - \omega_{uv}$. The TOM-based dissimilarity d^{ω}_{uv} is used as the input for average linkage hierarchical clustering. The output from hierarchical clustering is a dendrogram composed of branches and leaves. In a brain functional network, each leaf corresponds to a ROI. The hierarchical clustering algorithm groups the closest ROIs and forms the branches. By cutting the branches of the dendrogram, closely related ROIs are identified as a module. Among the several methods for cutting the branches of the dendrogram, the default used in the WGCNA framework is Dynamic Tree Cut from the R package `dynamicTreeCut`.

Once modules are identified, one can calculate an intramodular connectivity

$$\omega.in_u = \sum_{v \in M} \omega_{uv} \tag{3.3}$$

for each node u in its module M. Ref. [20] pointed out that intramodular connectivities $\omega.in$ may represent important features of the nodes (i.e. ROIs).

3.2.2 An adaptive association test based on the proportional odds model

Let $Y_i = 0, 1, 2$ denote the count of the minor allele for subject i for a given SNP of interest, then Y_i has $J = 3$ ordered categories. The logistic regression model cannot be applied in this situation, because it only allows the response variable to be binary. A popular choice for ordinal data is the proportional odds model (POM)[62], which we will briefly describe here.

Suppose subject i has p network connectivities denoted by $X_i = (x_{i1}, \ldots, x_{ip})$ and l covariates denoted by $Z_i = (z_{i1}, \ldots, z_{il})$. For the proportional odds model, we define the regression coefficients $\beta = (\beta_1, \ldots, \beta_p)'$ for the network connectivities and $\delta = (\delta_1, \ldots, \delta_l)'$, and a vector of intercepts $\alpha = (\alpha_0, \ldots, \alpha_{J-2})'$. The proportional odds model is

$$\text{logit}[Pr(Y_i \leq j)] = \alpha_j + Z_i\delta + X_i\beta, \qquad j = 0, 1. \tag{3.4}$$

The likelihood for equation Eq. 3.4 can be derived based on the multinomial distribution for the categorical variable Y_i, from which maximum likelihood estimates and statistical inference can be obtained as implemented in R package `MASS` or `VGAM`. However, numerical issues such as non-convergence arise when p, the dimension of β, is relatively large as compared to the sample size n.

Here we propose applying a class of tests that are applicable to the high-dimensional setting with $p > n$, from which an adaptive test is constructed to summarize information across the tests. No that most existing tests cannot be applied to the case $p > n$. To test the null hypothesis $H_0 : \beta = (\beta_1, \beta_2, \ldots, \beta_p)' = 0$, we can use the score vector derived in Ref. [21],

$$U_\beta = \sum_{i=1}^{n} \sum_{j=0}^{J-2} (1 - \hat{r}_{i(j-1)} - \hat{r}_{ij}) \cdot I(Y_i = j) \cdot X_i \tag{3.5}$$

where $\hat{r}_{ij} = \exp(\hat{\alpha} + Z_i\hat{\delta})/[1 + \exp(\hat{\alpha} + Z_i\hat{\delta})]$ comes from the fitted null model of Eq. 3.4 (i.e. with $\beta = 0$); $\hat{\alpha}$ and $\hat{\delta}$ are estimated by the `polr` function in the R package `MASS`. Let U_k denote the kth component of the score vector $U_\beta = (U_1, \ldots, U_p)'$. The SPU($\gamma$) test statistic is defined as

$$T_{SPU(\gamma)} = \sum_{k=1}^{p} U_k^\gamma, \tag{3.6}$$

where $\gamma \geq 1$ is an integer. As the parameter γ increases, a connectivity with a larger absolute value of the score gains a higher weight. In the extreme situation, when $\gamma \to \infty$ as an even integer, $SPU(\infty)$ takes only the maximum component of the score vector, i.e., $T_{SPU(\infty)} = \max_{k=1}^{p} |U_k|$.

The p-values of the SPU tests are computed by permuting the residuals from the null model B times, and the p-value can be calculated as

$$P_{SPU(\gamma)} = \frac{(\sum_{b=1}^{B} I[|T_{SPU(\gamma)}^{(b)}| \geq |T_{SPU(\gamma)}|] + 1)}{(B + 1)}, \tag{3.7}$$

where $T_{SPU(\gamma)}^{(b)}$ is the SPU(γ) statistic based on the bth set of permuted residuals. Since the value of γ that yields highest power cannot be determined a priori, an adaptive SPU (aSPU) test is introduced to combine the evidence across multiple SPU tests,

$$T_{aSPU} = \min_{\gamma \in \Gamma} P_{SPU(\gamma)}, \tag{3.8}$$

where $P_{SPU(\gamma)}$ is the p-value of $SPU(\gamma)$ test statistics and Γ is a set of integers for the power of aSPU test. In the numerical examples throughout this paper, we chose γ from the set $\Gamma = \{1, 2, \ldots, 8, \infty\}$. To calculate the p-value of T_{aSPU}, we can use the same permutation scheme as used for calculating the p-values of T_{SPU}'s. For each permuted residual set b, after calculating $T^{(b)}_{SPU(\gamma)}$ and its p-value $p^{(b)}_{\gamma} = (\sum_{b_1 \neq b} I[T^{(b_1)}_{SPU(\gamma)} \geq T^{(b)}_{SPU(\gamma)}] + 1)/B$. Then we can obtain $T^{(b)}_{aSPU} = \min_{\gamma \in \Gamma} p^{(b)}_{\gamma}$, and the p-value of T_{aSPU} is

$$P_{aSPU} = \frac{(\sum_{b=1}^{B} I[T^{(b)}_{aSPU} \leq T_{aSPU}] + 1)}{(B+1)}. \tag{3.9}$$

A step-wise procedure is used to gradually increase B if needed. We can start with $B = 10^3$ initially, then increase to $B = 10^5$ (or bigger) if a p-value is smaller than 5×10^{-3} (or smaller). The test is implemented in R package `POMaSPU` to be available on CRAN.

3.3 Results

3.3.1 ADNI Data

Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (`adni.loni.usc.edu`). We included all subjects from the normal and Alzheimer's disease (AD) groups in the ADNI data. We applied motion correction and global signal regression to reduce noises.

Here we used the power adjacency function $a_{uv} = power(s_{uv}, \beta) = |s_{uv}|^{\beta}$ (equation (3.1)). β was selected as the smallest β such that the scale-free model fitting R^2 was above a pre-set threshold 0.85.

3.3.2 Distinct modular structures in brain functional networks based on APOE4 SNP genotype scores

For the ADNI data, we grouped the subjects based on the APOE4 SNP (rs429358) minor allele counts $(0, 1, 2)$. APOE4 plays a major role in the pathogenesis of Alzheimer's disease [63, 64]. The APOE4 variant is a major risk factor for both early- and late-onset Alzheimer's disease [63, 64]. We removed those subjects with a missing rs429358 value, resulting in a total of 162 subjects. Among them, 73 subjects have no minor

allele at rs429358, whereas 67 subjects have one minor allele and 22 subjects have two. In order to establish possible modular structures in brain functional networks in the normal condition, we first applied the WGCNA framework to the rs-fMRI data of the control subjects only. Specifically, for each ROI, we concatenated the BOLD time series of all the control subjects, which were used to calculate the similarity between any two ROIs (i.e. the absolute value of Pearson's correlation between any two BOLD time series), then conducting the subsequent analyses in the WGCNA framework. At the end, we identified four modules based on the data from the control cohort (Figure 3.1).

Based on the modules identified, we continued to explore them for each APOE4 SNP genotype group. To measure the network connectivities, we used the correlation matrix, covariance matrix, and the topological overlap matrix (TOM). The rows and columns are ordered in the same way as in Figure 3.1. Distinct modular structures seem to be present in the correlation, covariance and TOM plots across the APOE4 genotype groups (Figure 3.2).

3.3.3 Adaptive testing for SNP-module associations

Using the APOE4 SNP (rs429358) minor allele counts as the response in a POM, we tested the association between the APOE4 SNP and the network connectivities. Covariates including age, gender and years of education were adjusted. Using the aSPU test, we found that the covariance matrix elements were marginally associated with the APOE4 SNP ($P = 0.033$, Table 3.1). We further decomposed the whole network connectivities into two exclusive subsets: connectivities within the four modules and those between the modules. Both the between-modular covariance and TOM were associated with the APOE4 SNP with $P < 0.05$.

Next we focused on the network connectivities in each individual module, and tested their association with the APOE4 SNP (Table 3.2). The network connectivities defined by the correlations in the yellow module showed evidence of association with the APOE4 SNP ($P = 0.017$). In addition, the network connectivities defined by covariance matrix elements in the blue and yellow modules were also associated with the APOE4 SNP ($P = 0.034, P = 0.011$).

Finally we tested for association between each module-specific intramodular connectivity $\omega.in$ and the APOE4 SNP. Only the yellow module showed a significant association

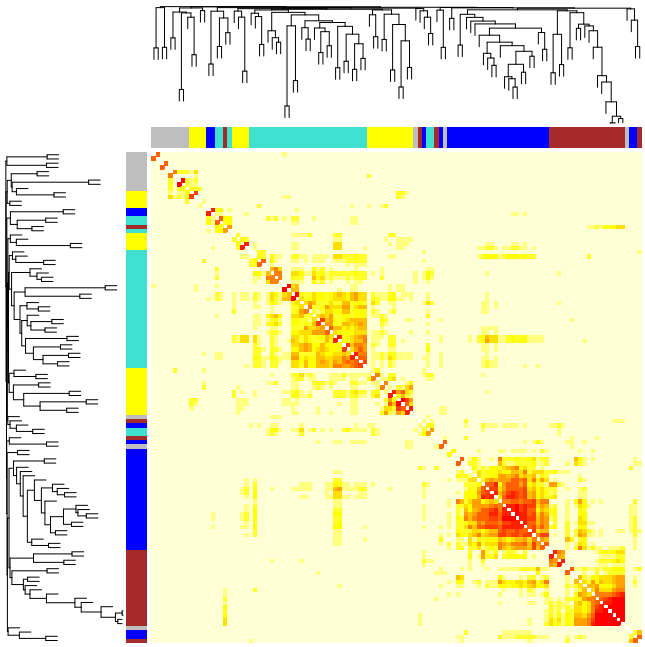

Figure 3.1: TOM plot of the whole brain functional network and its modules for normal subjects. The rows and columns are the ROIs, ordered by their distance in the tree.

with $P = 0.007$.

There are 30 and 19 ROIs in the blue and yellow modules, respectively. The ROIs identified in the yellow modules includes left/right sides of posterior cingulate cortex, angular gyrus, superior frontal cortex, middle frontal cortex, and inferior frontal cortex. For comparison, Ref. [21] identified 18 nodes related to the default mode network (DMN), including left/right sides of superior frontal cortex, medial prefrontal cortex, ventral anterior cingulate cortex, posterior cingulate cortex, parahipppocampal cortex, inferior parietal cortex, angular, middle temporal gyrus, and inferior temporal cortex [65, 13, 66]. We found that 15 ROIs in the yellow module are also related to the 18 nodes in the DMN. For example, the posterior cingulate cortex plays a pivotal role in the default mode network of the brain [67, 68]. The posterior cingulate cortex is linked

to cognitive functions such spatial memory, configural learning, and maintenance of discriminative avoidance learning and [69, 70]. It is shown in the DMN that Alzheimer's disease affects the posterior cingulate cortex [68]. Angular gyrus is another region found in both DMN and the yellow module. Loss of grey matter volume in angular gyrus has been associated with dementia and progression to Alzheimer's disease [71]. The association between the APOE4 variant and the network connectivity measures in the yellow module also uncovers some key brain regions in DMN that were found to be affected in Alzheimer's disease.

The ROIs in the blue module includes the left/right sides of hippocampus, lingual gyrus, cuneus, calcarine fissure and superior occipital gyrus, cerebellum and vermis. Hippocampus is well known for its key role in memory [72]. Hippocampal neuronal loss and structural change have been connected with Alzheimer's disease [73, 74]. Alzheimer's disease patients have also demonstrated neuronal and glial loss and structural changes in cerebellum and vermis [75]. Lingual gyrus, cuneus, calcarine fissure and superior occipital gyrus are located in the occipital lobe, which are mainly related to vision processing [76]. In addition, lingual gyrus plays an important role in the identification and recognition of words [77]. The association between the APOE4 SNP and the network connectivity measures may reflect the pathological changes of the brain functional network in Alzheimer's disease.

SNP Count = 0 SNP Count = 1 SNP Count = 2

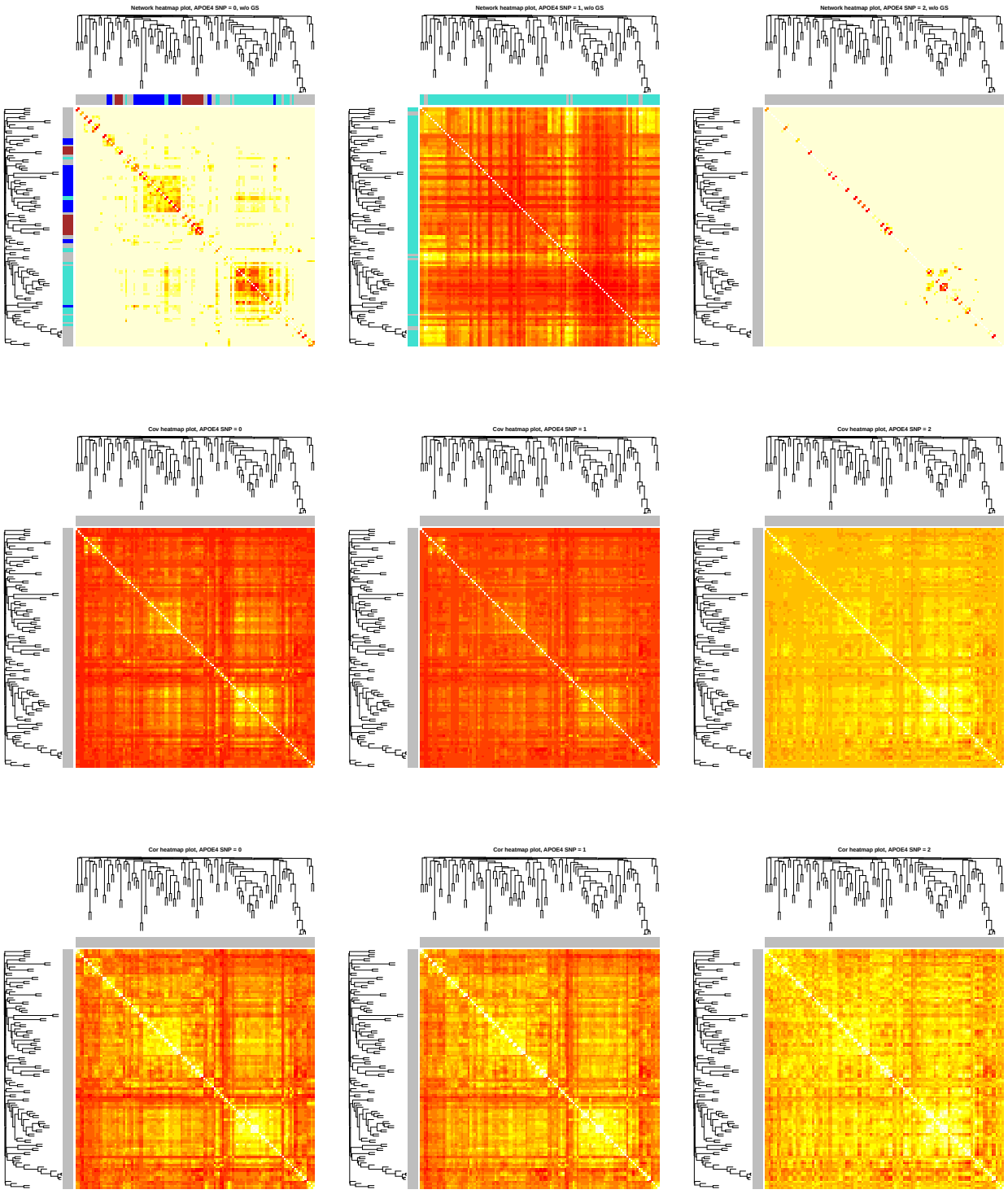

Figure 3.2: TOM plot (top), covariance matrix plot (middle) and correlation matrix plot (bottom) of the brain functional networks for the three genotype groups based on APOE4 SNP (rs429358) (with its minor allele counts equal to 0, 1 or 2 from left to right).

	Correlation			Covariance			TOM			Adjacency		
Test	All	W-mod	Btw-mod	All	W-mod	Btw-mod	All	W-mod	Btw-mod	All	W-mod	Btw-mod
SPU(1)	0.477	0.365	0.526	0.052	0.052	0.087	0.530	0.637	0.109	0.477	0.527	0.479
SPU(2)	0.161	0.154	0.207	0.012	0.016	0.008	0.099	0.250	0.014	0.477	0.515	0.487
SPU(3)	0.323	0.202	0.377	0.018	0.025	0.010	0.817	0.902	0.224	0.472	0.482	0.498
SPU(4)	0.150	0.122	0.197	0.019	0.009	0.004	0.325	0.424	0.172	0.463	0.434	0.516
SPU(5)	0.248	0.137	0.299	0.130	0.066	0.004	0.892	0.987	0.317	0.442	0.402	0.528
SPU(6)	0.141	0.101	0.209	0.120	0.008	0.003	0.444	0.533	0.330	0.416	0.365	0.554
SPU(7)	0.216	0.111	0.267	0.429	0.052	0.004	0.657	0.890	0.393	0.381	0.348	0.568
SPU(8)	0.137	0.089	0.225	0.188	0.009	0.004	0.498	0.603	0.410	0.348	0.334	0.583
SPU(∞)	0.122	0.079	0.356	0.210	0.009	0.007	0.463	0.706	0.655	0.263	0.239	0.460
aSPU	0.208	0.146	0.296	0.033	0.025	0.007	0.181	0.384	0.039	0.356	0.328	0.585

Table 3.1: P-values of the tests for SNP-whole network associations using the correlation, covariance, TOM or adjacency matrix elements as the network connectivity measure respectively. W-mod and Btw-mod stand for within-modular and between-modular, respectively.

Module	Test	Correlation		Covariance		TOM		Adjacency	
		W-mod	Btw-mod	W-mod	Btw-mod	W-mod	Btw-mod	W-mod	Btw-mod
Blue	SPU(1)	0.140	0.272	0.018	0.012	0.946	0.093	0.504	0.488
	SPU(2)	0.090	0.077	0.025	0.001	0.440	0.067	0.515	0.481
	SPU(3)	0.099	0.144	0.028	0.001	0.793	0.396	0.502	0.455
	SPU(4)	0.101	0.071	0.033	0.001	0.528	0.265	0.496	0.423
	SPU(8)	0.139	0.075	0.050	0.011	0.566	0.458	0.464	0.264
	SPU(∞)	0.267	0.070	0.069	0.015	0.571	0.554	0.458	0.180
	aSPU	0.160	0.119	0.034	0.003	0.654	0.141	0.581	0.244
Turquoise	aSPU	0.648	0.172	0.277	0.011	0.139	0.192	0.605	0.309
Brown	aSPU	0.260	0.182	0.016	0.015	0.459	0.219	0.249	0.327
Yellow	SPU(1)	0.040	0.500	0.005	0.084	0.083	0.172	0.357	0.510
	SPU(2)	0.024	0.219	0.005	0.012	0.060	0.155	0.323	0.509
	SPU(3)	0.020	0.350	0.005	0.015	0.147	0.458	0.297	0.514
	SPU(4)	0.016	0.220	0.010	0.006	0.188	0.445	0.262	0.522
	SPU(8)	0.008	0.271	0.100	0.004	0.317	0.690	0.177	0.500
	SPU(∞)	0.006	0.525	0.200	0.008	0.383	0.848	0.129	0.411
	aSPU	0.017	0.354	0.011	0.010	0.106	0.289	0.183	0.524

Table 3.2: P-values of the tests for SNP-individual network module associations using the correlation, covariance, TOM or adjacency matrix elements as the network connectivity measure.

3.3.4 GWAS scan with individual modules

We tested for associations of the SNPs across the whole genome with the functional connectivity measures in the yellow and blue modules respectively. For genotype data, we included all SNPs with a minor allele frequency (MAF) ≥ 0.05, genotyping rate $\geq 90\%$, and passing the Hardy-Weinberg equilibrium test with a p-value > 0.001. After filtering with the above criteria, we obtained 579,382 SNPs.

The genome-wide scan showed that among the SNPs associated with the network connectivities (measured by Pearson's correlation) in the yellow module, rs17114690 on chromosome 14 was the only SNP that had a p-value smaller than 10^{-3}. Three SNPs were founded to be associated with the network connectivities (correlations) in the blue module, with p-values smaller than 10^{-3}. They are located on chromosome 1 (rs7536105, rs11265187) and chromosome 2 (rs17498117). rs7536105 is located in the chromatin interactive region, while rs11265187 is located in the enhancer region of gene olfactory receptor family 10 subfamily J member 9 pseudogene (OR10J9P).

The genome-wide scan also identified 5 SNPs associated with the intramodular network connectivity $\omega.in$ for the yellow module, with $P < 10^{-5}$. They are located on chromosome 1 (rs6656071, rs12043216), chromosome 7 (rs1178127, rs12674460), and chromosome 13 (rs2819239). SNP rs1178127 is a missense variant in gene histone deacetylase 9 (HDAC9) [78], an important gene with function in transcriptional regulation and cell cycle in the Wnt signalling pathway.

3.4 Discussion

In this paper we adapted WGCNA for network construction and module detection to rs-fMRI data. Based on the identified modules, we also proposed applying a new adaptive association test for single SNP association with the connectivities of the whole network or its components in a proportional odds model. While the whole network was not associated, some module-based connectivities were significantly associated with the APOE4 SNP rs429358. Given the major role of APOE4 in the pathogenesis of Alzheimer's disease, our finding seems plausible, suggesting its possible use for genome-wide scans to detect SNP variants associated with altered brain networks and AD. Although none of the associations was highly or genome-wide significant, it was perhaps

due to a too small sample size; larger studies are needed. Our use of modules, with either various ROI-to-ROI connectivities (e.g. TOM in addition to standard correlations) or some module-based node measures (such as intramodular connectivity), not only may reduce the dimension and thus improve the statistical power, but also can enhance result interpretation, highlighting where is the association if any. In particular, we found that intramodular connectivities showed more significant associations with more SNPs, possibly due to their lower dimensions (i.e. p_1 in a module with p_1 ROIs as compared to $p_1(p_1-1)/2$ of ROI-to-ROI connectivities) and/or higher information contents.

The multiple traits used in this paper, including various network connectivity measures in the whole network or its various subcomponents, differ from most of the previous neuroimaging studies [79], in which the focus was on some direct measures on ROIs, not their connectivities as shown here. These phenotypes are often high dimensional with dimension exceeding the sample size. Many software packages cannot handle such a situation with $p > n$, which limits their use. The adaptive association test used in this paper can be applied to such high-dimensional traits. It can be a useful and powerful method for identifying associations between high-dimensional neuroimaging traits and SNPs. In this paper, we have focused on the study of the association between neuroimaging phenotypes and SNP genotype scores; however, other ordinal outcomes such as a disease status (e.g. normal, MCI and AD in the ADNI data) can be tested for their associations with neuroimaging and other endophenotypes.

Acknowledgment

This research was supported by NIH grants R01GM113250, R01HL105397 and R01HL116720, and by the Minnesota Supercomputing Institute.

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: Alzheimer's Association; Alzheimer's Drug Discovery Foundation; BioClinica, Inc.; Biogen Idec Inc.; Bristol-Myers Squibb Company; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and

Company; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; GE Healthcare; Innogenetics, N.V.; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research Development, LLC.; Johnson Johnson Pharmaceutical Research Development LLC.; Medpace, Inc.; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Synarc Inc.; and Takeda Pharmaceutical Company. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (`www.fnih.org`). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Disease Cooperative Study at the University of California, San Diego. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of California, Los Angeles. This research was also supported by NIH grants P30 AG010129 and K01 AG030514.

Chapter 4

An R package for estimation of directed acyclic graph

4.1 Introduction

A graph is composed of nodes and edges. The variables of interest are represented by the nodes in the graph, and an edge between two nodes indicates dependence between two variables. Graphical models are intuitive tools to describe dependence relationship between variables of interest. In addition, graphical models under distributional assumptions can be estimated and interpreted via statistical models. For example, the Gaussian graphical model exploits the conditional dependence relationship of the multivariate normal distribution ([80]).

The edges can be directed or undirected in a graphical model. An undirected graph contains information on the dependence relations between variables, but the direction of dependence cannot be specified without the direction on the edge. In contrast, direction of dependence can be inferred from directed graphs. Directed graphs are useful to reconstruct directional gene regulatory networks and SNP-gene regulatory networks. In particular, Directed acyclic graphs (DAGs) models are popular due to their simple interpretation of causal relationships. As indicated by its name, DAGs exclude directed circles among variables. However, the acyclicity constraints adds more difficulty to the problem of estimating DAGs.

Several methods have been proposed for estimating DAGs. For the low-dimensional

setting where the number of nodes is relatively small, a few methods have been proposed by either reducing the search space [22] or greedy search[23]. However, these methods struggles in the high-dimensional setting because the search space grows super exponentially in the number of nodes [24]. Another class of methods originated from the PC algorithm [25], which uses conditional independence relationship to delete recursive edges from an undirected graph. Kalisch and Bühlmann [25] adapted the PC algorithm for estimating the skeleton of DAGs in the high-dimensional setting. However, this approach is based on the strong faithfulness assumption that is restrictive to a small set of distributions.

Recently, Yuan et al. developed a new approach for estimation of DAGs in high-dimensional setting. In this article, we first introduced the method for estimation of directed acyclic Gaussian graph, and implemented this method as the R package gDAG. We also reviewed the graphical lasso and its extension as competing methods for estimating graphs. We compared the performance of gDAG with graphical lasso and other competing methods. We also demonstrated an application of the gDAG package to the gene expression and SNP data set by Webster et al. ([26]). We concluded this article with a brief discussion.

4.2 Methods

4.2.1 Estimation of directed acyclic graph

A DAG specifies the parent-child dependence relations of random vectors $(X_1, \ldots, X_p)$ with directed edges between nodes. The parents of X_j are the set of nodes with directional edges towards X_j in the graph. To estimate directional effects, the following structural model can be used, where

$$X_j = f_j(\mathrm{pa}_j, Z_j), \qquad j = 1, \ldots, p \tag{4.1}$$

If each f_j in (4.1) is linear, and each Z_j follows normal distribution, i.e., $Z_j \sim N(0, \sigma^2)$, then (4.1) becomes a Gaussian structural equation model,

$$X_j = \sum_{k \neq j} A_{jk} X_k + Z_j, \qquad j = 1, \ldots, p \tag{4.2}$$

where $A_{jk} = 0$ when node k is no in pa_j. Then unknown matrix $A_{p\times p}$ contains non-zero entries that the directional relationships among variables. To estimate A, we first specify the negative log-likelihood given data matrix $X_{n\times p}$. After dropping the constant terms, the negative log-likelihood becomes

$$l(A) = \frac{1}{2}\sum_{j=1}^{p}\sum_{i=1}^{n}(x_{ij} - \sum_{k\neq j} x_{ik}A_{jk})^2 \tag{4.3}$$

Constraints can be imposed to enforce the acyclicity and sparsity of the estimated graph, and the problem can be formulated as

$$\begin{aligned} &\min_{A} \quad l(A) \\ &\text{s.t.} \quad \sum_{j\neq k} I(A_{jk} \neq 0) \leq K \\ &\lambda_{ik} + I(j \neq k) - \lambda_{jk} \geq I(A_{ij} \neq 0); \qquad i,j,k = 1,\ldots,p, j \neq i \end{aligned} \tag{4.4}$$

where the first constraint in (4.4) imposes sparsity and second constraint in (4.4) guarantees acyclicity. The indicator functions $I(x)$ in the constraints can be approximated by the truncated L1 penalty $J_\tau(x) = \min(|x|/\tau, 1)$ (TLP, ([43])). $J_\tau(x)$ can be decomposed into a difference of two convex functions, $J_\tau(x) = S_1(x)S_2(x) = |x|/\tau - \max(|x|/\tau - 1, 0)$. $S_2(x)$ can be linearized by its majorization, then the original problem becomes

$$\begin{aligned} &\min_{A} \quad l(A) \\ &\text{s.t.} \quad \sum_{j\neq k} |A_{jk}| w_{ij}^{(m-1)} \leq Z^{(m-1)} \\ &\tau\lambda_{ik} + \tau I(j \neq k) - \tau\lambda_{jk} \geq |A_{ij}| w_{ij}^{(m-1)} + \tau(1 - w_{ij}^{(m-1)}); \quad i,j,k = 1,\ldots,p, j \neq i \end{aligned} \tag{4.5}$$

where $w_{ij}^{(m-1)} = I(\hat{A}_{ij}^{(m-1)} \leq \tau)$, $Z^{(m-1)} = \tau(K - \sum_{i\neq j}(1 - w_{ij}^{(m-1)}))$. We can solve this problem by alternative direction method of multipliers (ADMM) ([42]). The augmented Lagrangian is

$$\begin{aligned} L_\rho(A, B, \lambda, \epsilon, y, U) &= l(A) + \mu \sum_{i\neq j} |B_{ij}| w_{ij}^{(m-1)} + \rho/2 ||A - B + U||_F^2 \\ &+ \sum_k \sum_{i\neq j} \rho/2 \left(|B_{ij}| + w_{ij}^{(m-1)} + \tau(1 - w_{ij}^{(m-1)}) + \epsilon_{ijk} - \lambda_{ik} - \tau I(j \neq k) + \lambda_{jk} + y_{ijk} \right)^2 \end{aligned} \tag{4.6}$$

The computational algorithm is given as follows:

1. Initialization: Start with an initial estimate $\hat{A}^{(0)}$.

2. Iteration: At each iteration m, compute $\hat{A}^{(m)}$ by solving (4.6).

3. Convergence: Terminate when $l(\hat{A}^{(m-1)}) - l(\hat{A}^{(m)})$, where ϵ is the tolerance. We implement this algorithm as the `gDAG` package in `R`.

Additional continuous or discrete variables can be introduced as interventional variables to the DAG model. Let the $p \times p$ matrix A denote the adjacency matrix as in (4.4), where $A_{ij} \neq 0$ indicates an edge from node j to node i. The $p \times W$ matrix B encodes the intervention effect, and B_{iw} indicates the strength of intervention of X_w on Y_i. Placing A and B into the Gaussian structural equation models, we arrive at the following model,

$$Y_j = \sum_{k \neq j} A_{jk} Y_k + \sum_{w=1}^{W} B_{wj} X_w + Z_j, \qquad j = 1, \ldots, p \tag{4.7}$$

We can use similar approach to formulate the negative loglikelihood with regularization, and obtain the estimates of A and B via ADMM. In Section 4.5 we demonstrated an application of the gDAG package to a real data set using interventional models.

4.2.2 Review and extension of the graphical lasso

In this section we also review the graphical lasso method ([2]) for estimation of undirected Gaussian graph. Let $X_{n \times p} \sim N(\mu, \Sigma)$. To estimate the precision matrix $W = \Sigma^{-1}$, we only need to consider the terms in likelihood that contain W

$$\frac{n}{2} \log det(X) - \frac{1}{2} \sum_{j=1}^{n} (x_j - \mu)' W (x_j - \mu). \tag{4.8}$$

So we need to maximize

$$\frac{n}{2} \Big((\log det(W) - tr(\tilde{S}W) \Big) \tag{4.9}$$

with

$$\tilde{S} = \frac{\sum_{j=1}^{n} (x_j - \bar{x})(x_j - \bar{x})'}{n}. \tag{4.10}$$

If we impose a penalty on W, then we need to maximize

$$n \Big((\log det(W) - tr(\tilde{S}W) \Big) - p_\lambda(W) \tag{4.11}$$

For the graphical lasso, L_1-penalty is used to shrink the entries of the precision matrix W, that is,

$$p_\lambda(W) = \lambda||W||_1 \tag{4.12}$$

where λ_1 is a non-negative tuning parameter, $||W_i||$ is the sum of the absolute values of the entries of W_i. Boyd et al. ([42]) used ADMM to find the solution for the following problem,

$$\text{minimize} \left(-n(\log det(W) - tr(\tilde{S}W)) + \lambda \sum_{j=1}^{n} |w|_j \right) \tag{4.13}$$

We can apply ADMM to solve this lasso problem efficiently. To begin with, the augmented Lagrangian for the above problem (4.13) is given by

$$L_\eta W, Z, U) = -n \left(\log det(W - tr(\tilde{S}W \right) + \lambda \sum_{j=1}^{n} |z_j| + \frac{\eta}{2}||W - Z + U||_2^2 - \frac{\eta}{2}||U||_2^2 \tag{4.14}$$

Then the ADMM algorithm iteratively updates W, Z, and U iteratively as follows:

1. Initialization: Set $W = \mathbf{I}$, $Z = U = \mathbf{0}$.

2. Iteration: At each iteration m, update $W^{(m)}$, $Z^{(m)}$ and $U^{(m)}$:

(i) Update $W^{(m)}$ as the minimizer of

$$-n \left(\log det(W - tr(\tilde{S}W \right) + \frac{\eta}{2}||W - Z + U||_2^2. \tag{4.15}$$

with some algebra, we have the diagonal matrix $\tilde{W}$ with

$$\tilde{W}_{jj} = \frac{-\Lambda_{jj} + \sqrt{\Lambda_{jj}^2 + 4\eta/n}}{2\eta/n} \tag{4.16}$$

and $W^{(m)} = P\tilde{W}P^T$.

(ii) Update $Z^{(m)}$ as the miminzer of

$$\frac{\eta}{2}||Z^{(m)} - W^{(m)} + U^{(m)}||_2^2 + \lambda \sum_{j=1}^{n} |z_j| \tag{4.17}$$

The updating formula is given by elementwise soft-thresholding,

$$z_{ij}^{(m)} = S(w_{ij}^{(m)} + u_{ij}^{(m-1)}, \lambda/\eta) \tag{4.18}$$

where $S(x, c) = \text{sgn}(|x| - c)_+$ is the soft-thresholding operator.
(iii) Update $U^{(m)} = U^{(m-1)} + W^{(m)} - Z^{(m)}$.

3. Convergence: the above algorithm is repeated until desired tolerance criterion is met If we use the following non-convex penalty function for W,

$$p_\lambda(W) = \lambda \sum_{i,j} J_\tau(w_{ij}) \tag{4.19}$$

where $J_\tau(z)$ is the TLP,then for the non-convex optimization problem

$$\text{minimize} \left(-n(\log det(W) - tr(\tilde{S}W)) + \lambda \sum_{i,j}^{n} J_\tau(w_{ij}) \right) \tag{4.20}$$

we use DC decomposition $J_\tau(z) = |z| - \max(|z| - \tau, 0)$ to decompose the above objective function into $S_1(W) - S_2(W)$, where

$$S_1(W) = -n(\log det(W) - tr(\tilde{S}W)) + \lambda \sum_{i,j} |w_{ij}| \tag{4.21}$$

and

$$S_2(W) = \lambda \sum_{i,j} \max(|w_{ij}| - \tau, 0) \tag{4.22}$$

$S_2(W)$ can be iteratively approximated by its minorization $\lambda \sum_{i,j} I(|w_{ij}^{(m)}| > \tau)|w_{ij}|$, and the function to minimize in (4.20) can be approximated by

$$-n(\log det(W) - tr(\tilde{S}W)) + \lambda \sum_{i,j} I(|w_{ij}^{(m)}| \leq \tau)|w_{ij}| \tag{4.23}$$

We can apply ADMM to solve this lasso problem effciently.

4.3 A short tutorial for the gDAG package

We illustrate the installation and usage of the `gDAG` package with a short tutorial. The current `gDAG` package is built under the Unix system. In a Unix-like system, put the package tar ball into your working directory and open `R`. Use the following command to install the `gDAG` package,

```
install.packages("gDAG_1.0.tar.gz", repos = NULL, type="source")
```

After installation, we can call the functions in the `gDAG` package. We use a simple simulated data set from Model 1 in Section 4 as an example. The data set contains 1000 observations and 3 variables.

```
library(gDAG)
n = 1000
isim = 20
set.seed(482348026+isim*1e6)

v1 = rnorm(n,0,1)
v2 = rnorm(n,0,1)
v3 = v1 + v2 + rnorm(n,0,1)

X = cbind(v1,v2,v3)
```

We used the `cv.gDAG` function to estimate a DAG for the simulated data. The tuning parameters are selected via 5-fold cross-validation. The `Ahat` matrix in the resulting `cv.gDAG.out` object contains the estimated coefficient for each edge.

```
cv.gDAG.out = cv.gDAG(X=X,cv.fold=5, seed = NULL,
tau = seq(0.5,2,0.1), mu = c(1,5,10), rho = 0.01, tol = 1e-5)
cv.gDAG.out$Ahat
> cv.gDAG.out$Ahat
          [,1]      [,2]  [,3]
[1,] 0.0000000 0.0000000     0
[2,] 0.0000000 0.0000000     0
[3,] 0.9990683 0.9473243     0
```

We can use plots to visualize the estimated DAG. To do so, we will use the `Rgraphviz` package and convert the estimated coefficient matrix to a `graphNEL` class object. Then we can use the plot method in the `Rgraphviz` package to plot the estimated DAG (Figure 4.1).

4.4 Simulations

We considered two models, each having three nodes. Both models have two directed edges linking two nodes to the third one, but the directions are reversed. The two models can be specified as follows

$$\begin{aligned} X, Y &\overset{i.i.d.}{\sim} N(0,1) \\ Z &= X + Y + \epsilon, \epsilon \sim N(0,1) \end{aligned} \tag{4.24}$$

$$\begin{aligned} Z, \epsilon, &\overset{i.i.d.}{\sim} N(0,1) \\ X &= Z + \epsilon, \\ Y &= Z + \epsilon, \end{aligned} \tag{4.25}$$

We simulated 100 data sets with varying sample size $n = 100, 1000, 10000$ based on Model 1 and 2. We applied the `gDAG` package, the graphical lasso with either L_1 or TLP penalty to the simulated data sets. The tuning parameter $\lambda \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2, 5\}$ and $\tau \in \{0.01, 0.1, 1, 10\}$ were chosen based on 5-fold cross validation. Table 4.1 showed the percentage of non-zero elements in the estimated graph for Model 1. All three methods recovered the true edges with 100% percentage. False positive edges were present in the gDAG package occasionally. It is noteworthy that the two versions of the graphical lasso returned undirected graphs, therefore the matrices representing the estimated graphs are symmetric. In contrast, the estimated graph returned by the `gDAG` package contained directed edges and the matrices representing the estimated graphs are therefore asymmetric. For Model 2, all three methods correctly recovered the true edges with 100% percentage, but the false positive rates for the gDAG package decreased as the sample size of each simulated data set increased (Table 4.2). The two graphical lasso methods demonstrated high false positive rates, even when the sample size was large. Tables 4.3 and 4.4 showed the mean of the estimated parameters for Model 1 and Model 2. The values of the estimated parameters are close to those of the true model, suggesting that smaller penalty was selected. The plots of solution paths of the graphical lasso for Model 1 and Model 2 recognized the correct sparsity pattern (Figures 4.3 and 4.4), but in Model 2 the estimated parameters by the graphical lasso have small non-zero values for the false positive edge between node X and node Y. In addition, the

selected tuning paramter was small and thus encourages less penalty on the estimated parameters. This led to the high false positive rate for the graphical lasso in Model 2.

We also increased the regression coefficient of a directed edge in the previous two models, and thus dened Model 3 and Model 4 as follows,

$$\begin{aligned} X, Y &\overset{i.i.d.}{\sim} N(0,1) \\ Z &= 2X + Y + \epsilon, \epsilon \sim N(0,1) \end{aligned} \tag{4.26}$$

$$\begin{aligned} Z, \epsilon, &\overset{i.i.d.}{\sim} N(0,1) \\ X &= 2Z + \epsilon, \\ Y &= Z + \epsilon, \end{aligned} \tag{4.27}$$

The same simulation setting used in Model 1 and Model 2 were applied to Model 3 and Model 4. The `gDAG` package showed similar performance for Model 3 and Model 4 and recovered the directed edges in most cases for Model 3 (Table 4.5) and Model 4 (Table 4.6). False positive edges were present in some cases for Model 3. The graphical lasso methods also demonstrated similar performance in recovering the true edges. The false positive rate in Model 4 decreased as the sample size of each simulated data set increased. Since the true parameters had larger absolute values in Model 3 and Model 4, more penalty was applied and caused the parameter for the false edge to be shrunken to 0.

We also increased the number of nodes and illustrate the use of the `gDAG` package with Model 5 (Figure 4.5). The percentages for the true edges were highlighted for the `gDAG` pacakge. Even when the sample size is 100, the true positive rates are very high. The true positive rates reached 100% when the sample size is large (Tables 4.9 - 4.10). The false positive rates were also reasonably controlled. For such a complex model, it was more difficult to recognize the conditional dependence relationship between nodes by using the graphical lasso. Compared to the undirected graph estimated by the graphical lasso, the DAG estimated by the `gDAG` package is more intuitive and has better interpretability when the association between variables is directional.

4.5 An application to a study for Alzheimer's Disease

In this section, we demonstrate the use of the gDAG package with additional discrete variables, as discussed in . We applied the gDAG function to the data set used in Webster et al ([26]). Both gene expression profile and genotype information were reported for 187 healthy participants and 176 Alzheimers disease patients. We mapped the probes in the gene expression data set to the Alzheimers disease pathway in KEGG, resulting in 99 matched probes. We used two-sample t-test and found 40 probes were differentially expressed between the two groups. We added the genotype for SNP rs429358, which is mapped to the APOE4 gene and is discovered to be associated with AD. Lambert et al. ([81]) also identified several SNPs that are associated with AD, and weve chosen the genotype for SNP rs6656401 in our analysis. Both genes were significantly associated with the disease status in the data set we used here. The workflow is demonstrated in the example codes below. The input to the cv.gDAG function includes both the SNP minor allele counts and the gene expressions levels. The estimated graphs are plotted using igraph package. Figure 6 shows the estimated graphs by either the gDAG or the graphcial lasso for the pooled samples, the control group or the disease group. The graphs estimated by the gDAG clearly demonstrates the causal relationship from the SNP to its target genes via directed edges.

```
library(gDAG)

# load the data
load("/home/c/Desktop/gDAG/realexample.RData")

# fit the data with DAG model
out= cv.gDAG(X1=snp, X2=gene, cv.fold=5, seed = NULL,
tau = c(0.5,1.2,1.6), mu = c(1,5,10), rho = 0.01, tol = 1e-5)

# convert the fitted model to a graphNEL object for plotting
require(Rgraphviz)

g1hat <-as(abs(t(out$Ahat)), "graphNEL")
```

```
# use igraph package for plotting
require(igraph)

g1igraph = igraph.from.graphNEL(g1hat)
V(g1igraph)$name[c(1:2)] = c("rs6656401", "rs429358")
E(g1igraph)$color = "grey"

layout <- layout_on_sphere(g1igraph)
# normalize the coordinates of the nodes
layout1 <- layout.norm(layout)



layout1[2,c(1:2)] = c(-1,1)
layout1[1,c(1:2)] = c(1,-1)
plot(g1igraph, vertex.size=1, layout = layout1,
edge.arrow.size=.5, main = "Pooled Samples")
layout1[2,c(1:2)] = c(-1,1)
layout1[1,c(1:2)] = c(1,-1)
plot(g1igraph, vertex.size=1, layout = layout1,
edge.arrow.size=.5, main = "Pooled Samples"
```

4.6 Discussion

We implemented a recent method for estimating DAGs as the `gDAG` package in `R`. We demonstrated its use by a simple example in Section 3. In addition, we used simulated data sets to compare its performance to the graphical lasso. When the true model is a DAG, the `gDAG` package performs better than the graphical lasso. Not only is the true positive rate of discovering edges higher in the `gDAG` package in most cases, the estimated graph by the `gDAG` package also recovers the directions of dependence in the true model. In contrast, the graphical lasso and its extension only estimate an undirected graph, which does not contain any information on the direction of dependence.

The `gDAG` package can be used to estimate the graph when interventional variables are added. We demonstrated the use of the `gDAG` package in this situation via the application to a real data set. The estimated graph contains directed edges from SNP to genes, as well as other directed edges between genes. This conforms to the hierarchy of regulation in real life; the expression levels of genes are regulated by SNPs, but not vice versa. Though the graphical lasso is able to find conditional dependence in the real example, the direction of dependence cannot be inferred without other knowledge.

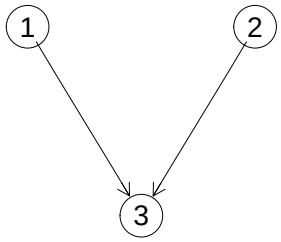

Figure 4.1: Plot of the estimated DAG for the simulated data set.

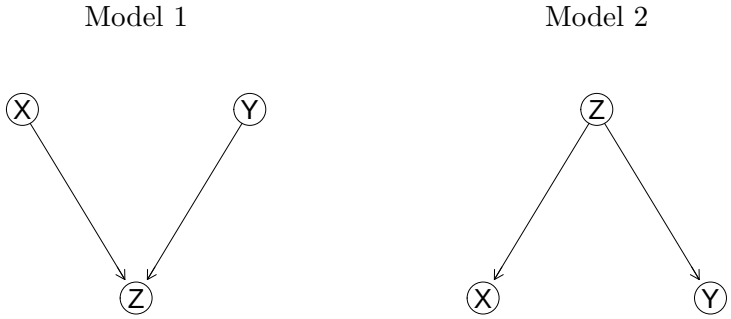

Figure 4.2: True DAG for Model 1 and Model 2.

		$n = 100$			$n = 1000$			$n = 10000$		
		X	Y	Z	X	Y	Z	X	Y	Z
DAG	X	0.00	0.50	0.00	0.00	0.48	0.00	0.00	0.46	0.00
	Y	0.50	0.00	0.00	0.52	0.00	0.00	0.54	0.00	0.00
	Z	1.00	1.00	0.00	1.00	1.00	0.00	1.00	1.00	0.00
gLasso	X	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Y	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Z	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TLPgLasso	X	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Y	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Z	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 4.1: % of non-zero elements for data set simulated from Model 1.

		$n = 100$			$n = 1000$			$n = 10000$		
		X	Y	Z	X	Y	Z	X	Y	Z
DAG	X	0.00	0.12	1.00	0.00	0.12	1.00	0.00	0.00	1.00
	Y	0.18	0.00	1.00	0.18	0.00	1.00	0.00	0.00	1.00
	Z	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
gLasso	X	1.00	0.90	1.00	1.00	0.87	1.00	1.00	0.89	1.00
	Y	0.90	1.00	1.00	0.86	1.00	1.00	0.89	1.00	1.00
	Z	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TLPgLasso	X	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Y	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Z	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 4.2: % of non-zero elements for data set simulated from Model 2.

		$n = 100$			$n = 1000$			$n = 10000$		
		X	Y	Z	X	Y	Z	X	Y	Z
DAG	X	0.00	-0.27	0.00	0.00	-0.25	0.00	0.00	-0.23	0.00
	Y	-0.26	0.00	0.00	-0.27	0.00	0.00	-0.27	0.00	0.00
	Z	1.00	1.00	0.00	1.00	1.01	0.00	1.00	1.00	0.00
gLasso	X	1.95	0.91	-0.95	1.99	1.00	-1.00	1.98	0.98	-0.99
	Y	0.91	1.96	-0.94	1.00	2.00	-1.00	0.98	1.98	-0.99
	Z	-0.95	-0.94	0.97	-1.00	-1.00	1.00	-0.99	-0.99	0.99
TLPgLasso	X	2.07	1.03	-1.04	2.01	1.02	-1.01	2.00	1.00	-1.00
	Y	1.03	2.08	-1.03	1.02	2.01	-1.01	1.00	2.00	-1.00
	Z	-1.04	-1.0	3 1.03	-1.01	-1.01	1.00	-1.00	-1.00	1.00

Table 4.3: Mean of the estimated parameters for data set simulated from Model 1.

		$n = 100$			$n = 1000$			$n = 10000$		
		X	Y	Z	X	Y	Z	X	Y	Z
DAG	X	0.00	0.01	1.01	0.00	0.01	1.01	0.00	0.00	1.00
	Y	-0.01	0.00	1.01	-0.01	0.00	1.01	0.00	0.00	1.00
	Z	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
gLasso	X	1.00	-0.03	-0.96	0.99	-0.01	-0.97	0.99	0.00	-0.99
	Y	-0.02	0.99	-0.94	-0.01	1.00	-0.99	0.00	1.00	-0.99
	Z	-0.96	-0.94	2.88	-0.97	-0.99	2.95	-0.99	-0.99	2.98
TLPgLasso	X	1.05	0.00	-1.06	1.00	-0.01	-0.99	1.00	0.00	-1.00
	Y	0.00	1.04	-1.04	-0.01	1.00	-1.00	0.00	1.00	-1.00
	Z	-1.06	-1.04	3.13	-0.99	-1.00	3.00	-1.00	-1.00	3.00

Table 4.4: Mean of the estimated parameters for data set simulated from Model 2.

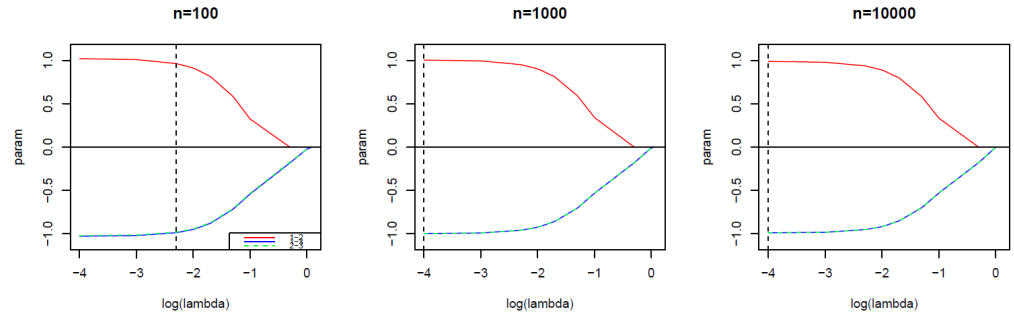

Figure 4.3: Solution paths of the graphical lasso for Model 1. The median of the selected tuning parameter was indicated by the dashed horizontal line.

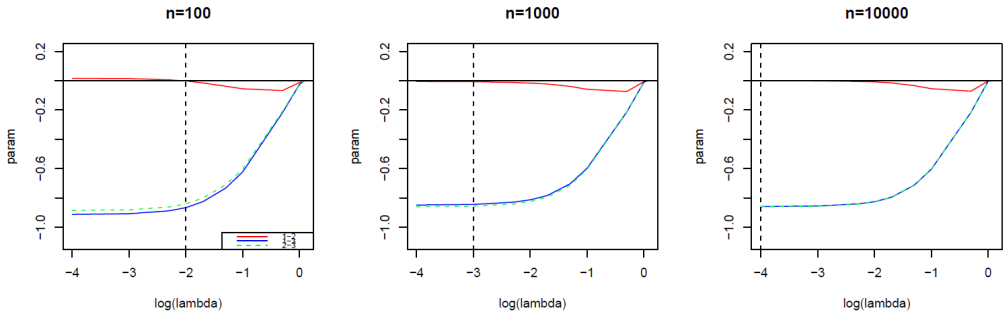

Figure 4.4: Solution paths of the graphical lasso for Model 2. The median of the selected tuning parameter was indicated by the dashed horizontal line.

		$n = 100$			$n = 1000$			$n = 10000$		
		X	Y	Z	X	Y	Z	X	Y	Z
DAG	X	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Y	0.66	0.00	0.00	0.71	0.00	0.00	0.70	0.00	0.00
	Z	1.00	0.88	0.00	1.00	0.86	0.00	1.00	0.85	0.00
gLasso	X	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Y	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Z	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TLPgLasso	X	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Y	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Z	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 4.5: % of non-zero elements for data set simulated from Model 3.

		$n = 100$			$n = 1000$			$n = 10000$		
		X	Y	Z	X	Y	Z	X	Y	Z
DAG	X	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
	Y	0.00	0.00	0.98	0.00	0.00	1.00	0.00	0.00	1.00
	Z	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
gLasso	X	1.00	0.34	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Y	0.34	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Z	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TLPgLasso	X	1.00	0.37	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Y	0.37	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Z	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 4.6: % of non-zero elements for data set simulated from Model 4.

		$n = 100$			$n = 1000$			$n = 10000$		
		X	Y	Z	X	Y	Z	X	Y	Z
DAG	X	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Y	-0.69	0.00	0.00	-0.73	0.00	0.00	-0.70	0.00	0.00
	Z	2.00	0.89	0.00	2.00	0.87	0.00	2.00	0.85	0.00
gLasso	X	4.94	1.93	-1.97	4.95	1.98	-1.98	4.92	1.96	-1.97
	Y	1.93	2.00	-0.97	1.98	1.99	-0.99	1.96	1.98	-0.98
	Z	-1.97	-0.97	0.99	-1.98	-0.99	0.99	-1.97	-0.98	0.99
TLPgLasso	X	5.15	2.04	-2.06	5.03	2.02	-2.01	5.00	2.00	-2.00
	Y	2.04	2.07	-1.02	2.02	2.01	-1.01	2.00	2.00	-1.00
	Z	-2.06	-1.02	1.03	-2.01 -	1.01	1.00	-2.00	-1.00	1.00

Table 4.7: Mean of the estimated parameters for data set simulated from Model 3.

		$n = 100$			$n = 1000$			$n = 10000$		
		X	Y	Z	X	Y	Z	X	Y	Z
DAG	X	0.00	0.00	2.01	0.00	0.00	1.99	0.00	0.00	2.00
	Y	0.00	0.00	1.00	0.00	0.00	0.99	0.00	0.00	1.00
	Z	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
gLasso	X	1.00	-0.02	-1.97	0.99	-0.01	-1.96	0.99	-0.00	-1.97
	Y	-0.02	1.01	-0.97	-0.01	1.00	-0.98	-0.00	1.00	-0.99
	Z	-1.97	-0.97	5.91	-1.96	-0.98	5.90	-1.97	-0.99	5.92
TLPgLasso	X	1.05	0.00	-2.10	1.00	-0.01	-1.99	1.00	0.00	-2.00
	Y	0.00	1.03	-1.04	-0.01	1.00	-1.00	0.00	1.00	-1.00
	Z	-2.10	-1.04	6.28	-1.99	-1.00	5.97	-2.00	-1.00	6.00

Table 4.8: Mean of the estimated parameters for data set simulated from Model 4.

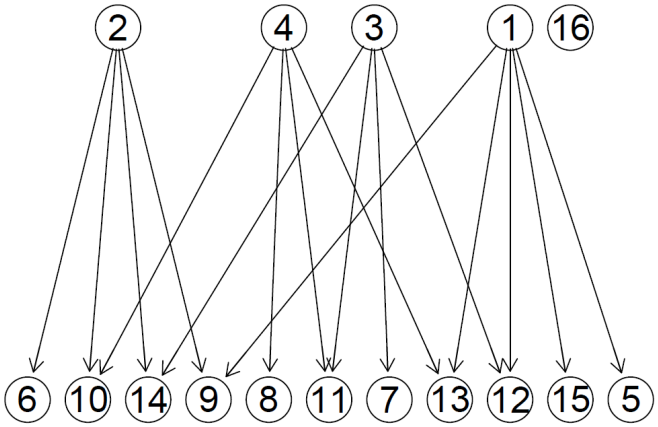

Figure 4.5: True DAG for Model 5.

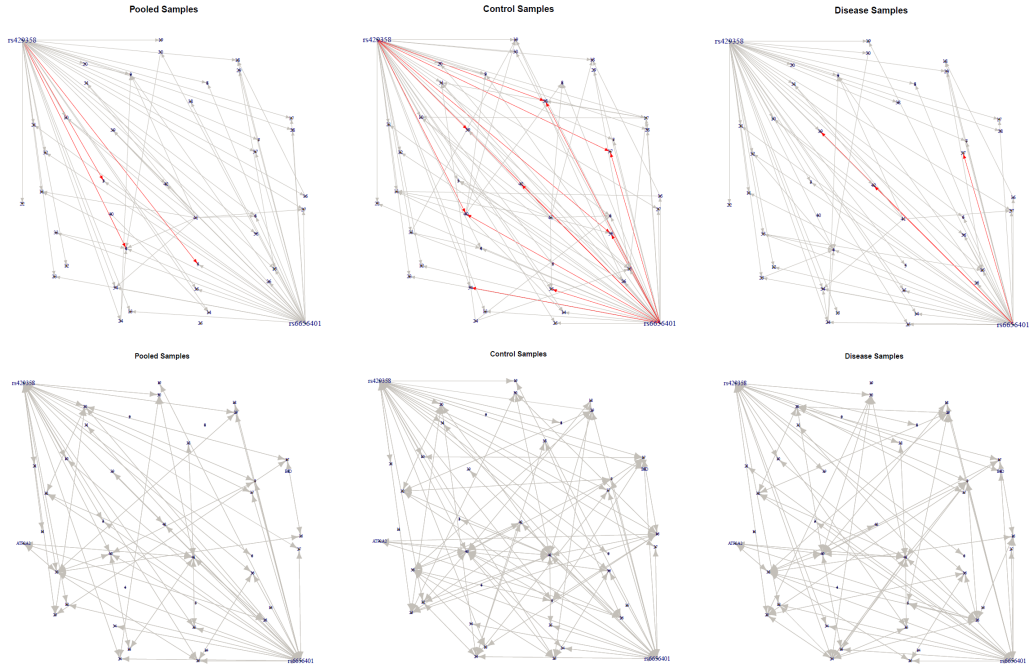

Figure 4.6: The estimated graph using the `gDAG` (top) or the graphical lasso (bottom) for the pooled samples, control cohort and the case cohort. The nodes for SNPs were relocated to corners for comparison.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	5	**0.98**	0.02	0.02	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	6	0.04	**0.98**	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	7	0.06	0.04	**0.99**	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	8	0.04	0.01	0.03	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	9	**0.94**	1.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	10	0.04	0.91	0.01	**0.97**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	11	0.04	0.03	1.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	12	**0.93**	0.03	**1.00**	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	13	**0.98**	0.03	0.01	**0.97**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	14	0.03	**0.98**	**0.96**	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	15	**0.95**	0.04	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	16	0.03	0.01	0.02	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
G	1	1.00	0.50	0.59	0.56	1.00	0.23	0.26	0.19	1.00	0.44	0.41	1.00	1.00	0.46	1.00	0.15
	2	0.50	1.00	0.69	0.62	0.26	1.00	0.21	0.21	1.00	1.00	0.46	0.49	0.54	1.00	0.21	0.13
	3	0.59	0.69	1.00	0.70	0.25	0.21	1.00	0.26	0.48	0.48	1.00	1.00	0.45	1.00	0.18	0.15
	4	0.56	0.62	0.70	1.00	0.18	0.25	0.22	1.00	0.49	1.00	1.00	0.39	1.00	0.50	0.24	0.13
	5	1.00	0.26	0.25	0.18	1.00	0.13	0.13	0.16	0.32	0.07	0.10	0.23	0.29	0.10	0.27	0.15
	6	0.23	1.00	0.21	0.25	0.13	1.00	0.06	0.09	0.23	0.33	0.08	0.10	0.15	0.21	0.13	0.15
	7	0.26	0.21	1.00	0.22	0.13	0.06	1.00	0.12	0.13	0.10	0.34	0.31	0.09	0.38	0.13	0.13
	8	0.20	0.21	0.26	1.00	0.16	0.09	0.12	1.00	0.15	0.37	0.24	0.08	0.26	0.12	0.12	0.14
	9	1.00	1.00	0.48	0.49	0.32	0.23	0.13	0.15	1.00	0.24	0.09	0.21	0.28	0.34	0.25	0.14
	10	0.44	1.00	0.48	1.00	0.07	0.33	0.10	0.37	0.24	1.00	0.18	0.10	0.18	0.25	0.09	0.11
	11	0.41	0.46	1.00	1.00	0.10	0.08	0.34	0.24	0.09	0.18	1.00	0.29	0.21	0.24	0.13	0.06
	12	1.00	0.49	1.00	0.39	0.23	0.10	0.31	0.08	0.21	0.10	0.28	1.00	0.26	0.22	0.29	0.11
	13	1.00	0.54	0.45	1.00	0.29	0.15	0.09	0.26	0.28	0.18	0.21	0.26	1.00	0.12	0.22	0.17
	14	0.46	1.00	1.00	0.50	0.10	0.21	0.38	0.12	0.34	0.25	0.24	0.22	0.12	1.00	0.08	0.12
	15	1.00	0.21	0.18	0.24	0.27	0.13	0.13	0.12	0.25	0.09	0.13	0.29	0.22	0.08	1.00	0.20
	16	0.15	0.13	0.15	0.13	0.15	0.15	0.13	0.14	0.14	0.11	0.06	0.11	0.17	0.12	0.20	1.00
T	1	1.00	0.93	0.89	0.94	1.00	0.65	0.69	0.67	1.00	0.70	0.71	1.00	1.00	0.79	1.00	0.75
	2	0.93	1.00	0.94	0.95	0.68	1.00	0.59	0.63	1.00	1.00	0.68	0.62	0.72	1.00	0.77	0.53
	3	0.89	0.94	1.00	0.96	0.59	0.75	1.00	0.64	0.67	0.67	1.00	1.00	0.62	1.00	0.61	0.61
	4	0.94	0.95	0.96	1.00	0.65	0.71	0.63	1.00	0.65	1.00	1.00	0.65	1.00	0.70	0.67	0.59
	5	1.00	0.68	0.59	0.65	1.00	0.47	0.39	0.46	0.48	0.39	0.42	0.40	0.41	0.31	0.37	0.41
	6	0.65	1.00	0.75	0.71	0.47	1.00	0.32	0.26	0.47	0.44	0.34	0.37	0.38	0.39	0.51	0.36
	7	0.69	0.59	1.00	0.63	0.39	0.32	1.00	0.39	0.44	0.37	0.43	0.42	0.43	0.55	0.36	0.45
	8	0.67	0.63	0.64	1.00	0.46	0.26	0.39	1.00	0.44	0.42	0.41	0.33	0.35	0.41	0.38	0.41
	9	1.00	1.00	0.67	0.65	0.48	0.47	0.44	0.44	1.00	0.47	0.35	0.43	0.44	0.45	0.40	0.38
	10	0.70	1.00	0.67	1.00	0.39	0.44	0.37	0.42	0.47	1.00	0.48	0.37	0.31	0.45	0.37	0.45
	11	0.71	0.68	1.00	1.00	0.42	0.34	0.43	0.41	0.35	0.48	1.00	0.37	0.42	0.35	0.41	0.40
	12	1.00	0.62	1.00	0.65	0.40	0.37	0.42	0.33	0.43	0.37	0.37	1.00	0.41	0.33	0.50	0.44
	13	1.00	0.72	0.62	1.00	0.41	0.38	0.43	0.35	0.44	0.31	0.42	0.41	1.00	0.37	0.40	0.49
	14	0.79	1.00	1.00	0.70	0.31	0.39	0.55	0.41	0.45	0.45	0.35	0.33	0.37	1.00	0.38	0.39
	15	1.00	0.77	0.61	0.67	0.37	0.51	0.36	0.38	0.40	0.37	0.41	0.50	0.40	0.38	1.00	0.48
	16	0.75	0.53	0.61	0.59	0.41	0.36	0.45	0.41	0.38	0.45	0.40	0.44	0.49	0.39	0.48	1.00

Table 4.9: % of non-zero elements for the estimated graphs from Model 5. Sample size $n = 100$. D stands for the gDAG package, G stands for the graphical lasso, and T stands for the graphical lasso with TLP. Non-zero percentage of the true edges were highlighted for the gDAG package.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
D	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	5	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	6	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	7	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	8	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	9	**1.00**	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	10	0.00	**1.00**	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	11	0.00	0.00	**1.00**	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	12	**1.00**	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	13	**1.00**	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	14	0.00	**1.00**	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	15	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
G	1	1.00	1.00	1.00	1.00	1.00	0.07	0.08	0.06	1.00	0.16	0.08	1.00	1.00	0.16	1.00	0.07	
	2	1.00	1.00	1.00	1.00	0.04	1.00	0.08	0.07	1.00	1.00	0.21	0.13	0.11	1.00	0.07	0.05	
	3	1.00	1.00	1.00	1.00	0.03	0.04	1.00	0.08	0.23	0.14	1.00	1.00	0.09	1.00	0.01	0.05	
	4	1.00	1.00	1.00	1.00	0.05	0.08	0.04	1.00	0.13	1.00	1.00	0.09	1.00	0.13	0.05	0.04	
	5	1.00	0.04	0.03	0.05	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	6	0.07	1.00	0.04	0.08	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	7	0.08	0.08	1.00	0.04	0.00	0.00	1.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	8	0.06	0.07	0.08	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	
	9	1.00	1.00	0.23	0.13	0.00	0.00	0.01	0.00	1.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	
	10	0.16	1.00	0.14	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.01	0.00	0.00	0.00	0.01	0.00	
	11	0.08	0.21	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.01	1.00	0.00	0.00	0.00	0.00	0.00	
	12	1.00	0.13	1.00	0.09	0.00	0.00	0.00	0.00	0.01	0.00	0.00	1.00	0.00	0.00	0.00	0.00	
	13	1.00	0.11	0.09	1.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	
	14	0.16	1.00	1.00	0.13	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	1.00	0.01	0.00	
	15	1.00	0.07	0.01	0.05	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	1.00	0.00	
	16	0.07	0.05	0.05	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	
T	1	1.00	1.00	1.00	1.00	1.00	0.16	0.20	0.15	1.00	0.23	0.17	1.00	1.00	0.22	1.00	0.18	
	2	1.00	1.00	1.00	1.00	0.09	1.00	0.16	0.15	1.00	1.00	0.24	0.14	0.18	1.00	0.14	0.20	
	3	1.00	1.00	1.00	1.00	0.07	0.15	1.00	0.21	0.19	0.20	1.00	1.00	0.13	1.00	0.10	0.10	
	4	1.00	1.00	1.00	1.00	0.12	0.15	0.08	1.00	0.16	1.00	1.00	0.14	1.00	0.14	0.10	0.12	
	5	1.00	0.09	0.07	0.12	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	6	0.16	1.00	0.15	0.15	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	
	7	0.20	0.16	1.00	0.08	0.00	0.00	1.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	8	0.15	0.15	0.21	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	
	9	1.00	1.00	0.19	0.16	0.00	0.00	0.01	0.00	1.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00	
	10	0.23	1.00	0.20	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.01	0.00	0.00	0.00	0.01	0.01	
	11	0.17	0.24	1.00	1.00	0.00	0.00	0.00	0.00	0.01	0.01	1.00	0.00	0.00	0.00	0.01	0.00	
	12	1.00	0.14	1.00	0.14	0.00	0.01	0.00	0.00	0.01	0.00	0.00	1.00	0.00	0.00	0.00	0.00	
	13	1.00	0.18	0.13	1.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	1.00	0.01	0.00	0.00	
	14	0.22	1.00	1.00	0.14	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	1.00	0.01	0.00	
	15	1.00	0.14	0.10	0.10	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.01	1.00	0.00	
	16	0.18	0.20	0.10	0.12	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	1.00	

Table 4.10: % of non-zero elements for the estimated graphs from Model 5. Sample size $n = 1000$. D stands for the gDAG package, G stands for the graphical lasso, and T stands for the graphical lasso with TLP. Non-zero percentage of the true edges were highlighted for the gDAG package.

References

[1] Roel GW Verhaak, Katherine A Hoadley, Elizabeth Purdom, Victoria Wang, Yuan Qi, Matthew D Wilkerson, C Ryan Miller, Li Ding, Todd Golub, Jill P Mesirov, et al. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in *PDGFRA*, *IDH1*, *EGFR*, and *NF1*. *Cancer cell*, 17(1):98–110, 2010.

[2] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[3] Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

[4] Sophie Achard, Raymond Salvador, Brandon Whitcher, John Suckling, and ED Bullmore. A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. *Journal of Neuroscience*, 26(1):63–72, 2006.

[5] Jonathan D Power, Alexander L Cohen, Steven M Nelson, Gagan S Wig, Kelly Anne Barnes, Jessica A Church, Alecia C Vogel, Timothy O Laumann, Fran M Miezin, Bradley L Schlaggar, et al. Functional network organization of the human brain. *Neuron*, 72(4):665–678, 2011.

[6] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p*) models for social networks. *Social networks*, 29(2):173–191, 2007.

[7] Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[8] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, pages 1436–1462, 2006.

[9] Yunzhang Zhu, Xiaotong Shen, and Wei Pan. Structural pursuit over multiple undirected graphs. *Journal of the American Statistical Association*, 109(508):1683–1696, 2014.

[10] Arthur P Dempster, Nan M Laird, Donald B Rubin, et al. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977.

[11] Wei Pan and Xiaotong Shen. Penalized model-based clustering with application to variable selection. *The Journal of Machine Learning Research*, 8:1145–1164, 2007.

[12] Hui Zhou, Wei Pan, and Xiaotong Shen. Penalized model-based clustering with unconstrained covariance matrices. *Electronic journal of statistics*, 3:1473, 2009.

[13] Lucina Q Uddin, AM Clare Kelly, Bharat B Biswal, F Xavier Castellanos, and Michael P Milham. Functional connectivity of default mode network components: correlation, anticorrelation, and causality. *Human Brain Mapping*, 30(2):625–637, 2009.

[14] Martijn P Van Den Heuvel and Hilleke E Hulshoff Pol. Exploring the brain network: a review on resting-state fmri functional connectivity. *European neuropsychopharmacology*, 20(8):519–534, 2010.

[15] Andrew Zalesky, Alex Fornito, Marc L Seal, Luca Cocchi, Carl-Fredrik Westin, Edward T Bullmore, Gary F Egan, and Christos Pantelis. Disrupted axonal fiber connectivity in schizophrenia. *Biological Psychiatry*, 69(1):80–89, 2011.

[16] Mary-Ellen Lynall, Danielle S Bassett, Robert Kerwin, Peter J McKenna, Manfred Kitzbichler, Ulrich Muller, and Ed Bullmore. Functional connectivity and brain networks in schizophrenia. *Journal of Neuroscience*, 30(28):9477–9487, 2010.

[17] Matthew K Belmonte, Greg Allen, Andrea Beckel-Mitchener, Lisa M Boulanger, Ruth A Carper, and Sara J Webb. Autism and abnormal development of brain connectivity. *The Journal of Neuroscience*, 24(42):9228–9231, 2004.

[18] Pedro A Valdés-Sosa, Jose M Sánchez-Bornot, Agustín Lage-Castellanos, Mayrim Vega-Hernández, Jorge Bosch-Bayard, Lester Melie-García, and Erick Canales-Rodríguez. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1457):969–981, 2005.

[19] Junghi Kim, Jeffrey R Wozniak, Bryon A Mueller, and Wei Pan. Testing group differences in brain functional connectivity: using correlations or partial correlations? *Brain connectivity*, 5(4):214–231, 2015.

[20] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4(1), 2005.

[21] Junghi Kim and Wei Pan. Adaptive testing for multiple traits in a proportional odds model with applications to detect snp-brain network associations. *Genetic Epidemiology*, 41(3):259–277, 2017.

[22] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[23] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

[24] Robert W Robinson. Counting unlabeled acyclic digraphs. *Combinatorial mathematics V*, 622(1977):28–43, 1977.

[25] Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(Mar):613–636, 2007.

[26] Jennifer A Webster, J Raphael Gibbs, Jennifer Clarke, Monika Ray, Weixiong Zhang, Peter Holmans, Kristen Rohrer, Alice Zhao, Lauren Marlowe, Mona Kaleem, et al. Genetic control of human brain transcript expression in alzheimer disease. *The American Journal of Human Genetics*, 84(4):445–458, 2009.

[27] Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 2011.

[28] Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[29] Chris Fraley and Adrian E Raftery. Mclust version 3: an r package for normal mixture modeling and model-based clustering. Technical report, WASHINGTON UNIV SEATTLE DEPT OF STATISTICS, 2006.

[30] Sijian Wang and Ji Zhu. Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics*, 64(2):440–448, 2008.

[31] Benhuai Xie, Wei Pan, and Xiaotong Shen. Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables. *Electronic journal of statistics*, 2:168, 2008.

[32] Steven M Hill and Sach Mukherjee. Network-based clustering with mixtures of l1-penalized gaussian graphical models: an empirical investigation. *arXiv preprint arXiv:1301.2194*, 2013.

[33] Meng-Yun Wu, Dao-Qing Dai, Xiao-Fei Zhang, and Yuan Zhu. Cancer subtype discovery and biomarker identification via a new robust network clustering algorithm. *PloS one*, 8(6):e66256, 2013.

[34] Nir Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, 2004.

[35] Adrian Dobra, Chris Hans, Beatrix Jones, Joseph R Nevins, Guang Yao, and Mike West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1):196–212, 2004.

[36] Donatello Telesca, Peter Müller, Steven M Kornblau, Marc A Suchard, and Yuan Ji. Modeling protein expression and protein signaling pathways. *Journal of the American Statistical Association*, 107(500):1372–1384, 2012.

[37] Mladen Kolar, Han Liu, and Eric P Xing. Graph estimation from multi-attribute data. *Journal of Machine Learning Research*, 15:1713, 2014.

[38] Shuai Huang, Jing Li, Liang Sun, Jieping Ye, Adam Fleisher, Teresa Wu, Kewei Chen, Eric Reiman, Alzheimer's Disease NeuroImaging Initiative, et al. Learning brain connectivity of alzheimer's disease by sparse inverse covariance estimation. *NeuroImage*, 50(3):935–949, 2010.

[39] Christine Peterson, Francesco C Stingo, and Marina Vannucci. Bayesian inference of multiple gaussian graphical models. *Journal of the American Statistical Association*, 110(509):159–174, 2015.

[40] Huitong Qiu, Fang Han, Han Liu, and Brian Caffo. Joint estimation of multiple graphical models from high dimensional time series. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(2):487–504, 2016.

[41] Karthik Mohan, Palma London, Maryam Fazel, Daniela Witten, and Su-In Lee. Node-based learning of multiple gaussian graphical models. *The Journal of Machine Learning Research*, 15(1):445–488, 2014.

[42] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[43] Xiaotong Shen, Wei Pan, and Yunzhang Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497):223–232, 2012.

[44] Cameron W Brennan, Roel GW Verhaak, Aaron McKenna, Benito Campos, Houtan Noushmehr, Sofie R Salama, Siyuan Zheng, Debyani Chakravarty, J Zachary Sanborn, Samuel H Berman, et al. The somatic genomic landscape of glioblastoma. *Cell*, 155(2):462–477, 2013.

[45] Roger McLendon, Allan Friedman, Darrell Bigner, Erwin G Van Meir, Daniel J Brat, Gena M Mastrogianakis, Jeffrey J Olson, Tom Mikkelsen, Norman Lehman, Ken Aldape, et al. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, 2008.

[46] Lewis C Cantley and Benjamin G Neel. New insights into tumor suppression: Pten suppresses tumor formation by restraining the phosphoinositide 3-kinase/akt pathway. *Proceedings of the National Academy of Sciences*, 96(8):4240–4245, 1999.

[47] Yoshitaka Narita, Motoo Nagane, Kazuhiko Mishima, HJ Su Huang, Frank B Furnari, and Webster K Cavenee. Mutant epidermal growth factor receptor signaling down-regulates p27 through activation of the phosphatidylinositol 3-kinase/akt pathway in glioblastomas. *Cancer research*, 62(22):6764–6769, 2002.

[48] Ghayda M Mirzaa, David A Parry, Andrew E Fry, Kristin A Giamanco, Jeremy Schwartzentruber, Megan Vanstone, Clare V Logan, Nicola Roberts, Colin A Johnson, Shawn Singh, et al. De novo ccnd2 mutations leading to stabilization of cyclin d2 cause megalencephaly-polymicrogyria-polydactyly-hydrocephalus syndrome. *Nature genetics*, 46(5):510–515, 2014.

[49] Zorbey Turkalp, Jason Karamchandani, and Sunit Das. Idh mutation in glioma: new insights and promises for the future. *JAMA neurology*, 71(10):1319–1325, 2014.

[50] Xiang Liu and Zhi-Qiang Ling. Role of isocitrate dehydrogenase 1/2 (idh 1/2) gene mutations in human tumors. *Histology and histopathology*, 30(10):1155–1160, 2015.

[51] Matija Snuderl, Joanna Triscott, Paul A Northcott, Helen A Shih, Esther Kong, Hayley Robinson, Sandra E Dunn, A John Iafrate, and Stephen Yip. Deep sequencing identifies idh1 r132s mutation in adult medulloblastoma. *Journal of Clinical Oncology*, 33(6):e27–e31, 2014.

[52] Geoffrey J McLachlan. On bootstrapping the likelihood ratio test stastistic for the number of components in a normal mixture. *Applied statistics*, pages 318–324, 1987.

[53] Joel H Reynolds and William D Templin. Comparing mixture estimates by parametric bootstrapping likelihood ratios. *Journal of Agricultural, Biological, and Environmental Statistics*, 9(1):57, 2004.

[54] Yvette I Sheline and Marcus E Raichle. Resting state functional connectivity in preclinical alzheimers disease. *Biological Psychiatry*, 74(5):340–347, 2013.

[55] Kaustubh Supekar, Vinod Menon, Daniel Rubin, Mark Musen, and Michael D Greicius. Network analysis of intrinsic functional brain connectivity in alzheimer's disease. *PLoS Computational Biology*, 4(6):e1000100, 2008.

[56] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[57] Claus C Hilgetag and Alexandros Goulas. Is the brain really a small-world network? *Brain Structure and Function*, pages 1–6, 2015.

[58] Junghi Kim, Wei Pan, and the Alzheimer's Disease Neuroimaging Initiative. Highly adaptive tests for group differences in brain functional connectivity. *NeuroImage: Clinical*, 9:625–639, 2015.

[59] Lingxue Zhu, Jing Lei, Bernie Devlin, and Kathryn Roeder. Testing high dimensional differential matrices, with application to detecting schizophrenia risk genes. *arXiv preprint arXiv:1606.00252*, 2016.

[60] Hyekyoung Lee, Hyejin Kang, Moo K Chung, Bung-Nyun Kim, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE transactions on medical imaging*, 31(12):2267–2277, 2012.

[61] Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555, 2002.

[62] Peter McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 109–142, 1980.

[63] Jungsu Kim, Jacob M Basak, and David M Holtzman. The role of apolipoprotein e in alzheimer's disease. *Neuron*, 63(3):287–303, 2009.

[64] Emmanuelle Genin, Didier Hannequin, David Wallon, Kristel Sleegers, Mikko Hiltunen, Onofre Combarros, María Jesús Bullido, Sebastiaan Engelborghs, Peter De Deyn, Claudine Berr, et al. Apoe and alzheimer disease: a major gene with semi-dominant inheritance. *Molecular Psychiatry*, 16(9):903–907, 2011.

[65] Michael D Greicius, Gaurav Srivastava, Allan L Reiss, and Vinod Menon. Default-mode network activity distinguishes alzheimer's disease from healthy aging: evidence from functional mri. *Proceedings of the National Academy of Sciences of the United States of America*, 101(13):4637–4642, 2004.

[66] Susanne Passow, Karsten Specht, Tom Christian Adamsen, Martin Biermann, Njål Brekke, Alexander Richard Craven, Lars Ersland, Renate Grüner, Nina Kleven-Madsen, Ole-Heine Kvernenes, et al. Default-mode network functional connectivity is closely related to metabolic activity. *Human Brain Mapping*, 36(6):2027–2038, 2015.

[67] Peter Fransson and Guillaume Marrelec. The precuneus/posterior cingulate cortex plays a pivotal role in the default mode network: Evidence from a partial correlation network analysis. *Neuroimage*, 42(3):1178–1184, 2008.

[68] Randy L Buckner, Jessica R Andrews-Hanna, and Daniel L Schacter. The brain's default network. *Annals of the New York Academy of Sciences*, 1124(1):1–38, 2008.

[69] Richard J Maddock, Amy S Garrett, and Michael H Buonocore. Remembering familiar people: the posterior cingulate cortex and autobiographical memory retrieval. *Neuroscience*, 104(3):667–676, 2001.

[70] Robert Leech and David J Sharp. The role of the posterior cingulate cortex in cognition and disease. *Brain*, 137(1):12–32, 2014.

[71] G Karas, J Sluimer, R Goekoop, W Van Der Flier, SARB Rombouts, H Vrenken, P Scheltens, N Fox, and F Barkhof. Amnestic mild cognitive impairment: structural mr imaging findings predictive of conversion to alzheimer disease. *American Journal of Neuroradiology*, 29(5):944–949, 2008.

[72] Larry R Squire. Memory and the hippocampus: a synthesis from findings with rats, monkeys, and humans. *Psychological Review*, 99(2):195, 1992.

[73] Bradley T Hyman, Gary W Van Hoesen, Antonio R Damasio, and Clifford L Barnes. Alzheimer's disease: cell-specific pathology isolates the hippocampal formation. *Science*, 225(4667):1168–1170, 1984.

[74] Mark J West, Paul D Coleman, Dorothy G Flood, and Juan C Troncoso. Differences in the pattern of hippocampal neuronal loss in normal ageing and alzheimer's disease. *The Lancet*, 344(8925):769–772, 1994.

[75] Martin Sjöbeck and Elisabet Englund. Alzheimers disease and the cerebellum: a morphologic study on neuronal and glial changes. *Dementia and Geriatric Cognitive Disorders*, 12(3):211–218, 2001.

[76] Rafael Malach, JB Reppas, RR Benson, KK Kwong, H Jiang, WA Kennedy, PJ Ledden, TJ Brady, BR Rosen, and RB Tootell. Object-related activity revealed by functional magnetic resonance imaging in human occipital cortex. *Proceedings of the National Academy of Sciences*, 92(18):8135–8139, 1995.

[77] Andrea Mechelli, Glyn W Humphreys, Kate Mayall, Andrew Olson, and Cathy J Price. Differential effects of word length and visual contrast in the fusiform and lingual gyri during. *Proceedings of the Royal Society of London B: Biological Sciences*, 267(1455):1909–1913, 2000.

[78] Ceres Fernandez-Rozadilla, Luisa De Castro, Juan Clofent, Alejandro Brea-Fernandez, Xavier Bessa, Anna Abuli, Montserrat Andreu, Rodrigo Jover, Rosa Xicola, Xavier Llor, et al. Single nucleotide polymorphisms in the wnt and bmp pathways and colorectal cancer risk in a spanish cohort. *PLoS One*, 5(9):e12673, 2010.

[79] Li Shen, Sungeun Kim, Shannon L Risacher, Kwangsik Nho, Shanker Swaminathan, John D West, Tatiana Foroud, Nathan Pankratz, Jason H Moore, Chantel D Sloan, Matthew J Huentelman, David W. Craig, Bryan M. DeChairo, Steven G. Potkin, Clifford R. Jack Jr, Michael W. Weiner, Andrew J. Saykin, and the Alzheimer's Disease Neuroimaging Initiative. Whole genome association study of brain-wide imaging phenotypes for identifying quantitative trait loci in mci and ad: A study of the adni cohort. *Neuroimage*, 53(3):1051–1063, 2010.

[80] Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[81] Jean-Charles Lambert, Carla A Ibrahim-Verbaas, Denise Harold, Adam C Naj, Rebecca Sims, Céline Bellenguez, Gyungah Jun, Anita L DeStefano, Joshua C Bis, Gary W Beecham, et al. Meta-analysis of 74,046 individuals identifies 11 new susceptibility loci for alzheimer's disease. *Nature genetics*, 45(12):1452–1458, 2013.