GRAPHICAL MODELS FOR HIGH DIMENSIONAL GENOMIC DATA

Min Jin Ha

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Biostatistics.

Chapel Hill
2013

Approved by:

Dr. Wei Sun
Dr. Joseph G. Ibrahim
Dr. Fred A. Wright
Dr. Michael G. Hudgens
Dr. William Valdar

Abstract

MIN JIN HA: Graphical Models for High Dimensional Genomic Data
(Under the direction of Dr. Wei Sun)

Graphical models study the relations among a set of random variables. In a graph, vertices represent variables and edges capture relations among the variables. We have developed three statistical methods for graphical model construction using high dimensional genomic data.

We first focus on estimating a high-dimensional partial correlation matrix. It is estimated by ridge penalty followed by hypothesis testing. The null distribution of the test statistics derived from penalized partial correlation estimates has not been established. We address this challenge by estimating the null distribution from the empirical distribution of the test statistics of all the penalized partial correlation estimates. The performance of our method is systematically evaluated in simulation and application studies.

Next, we consider estimating Directed Acyclic Graph (DAG) models for multivariate Gaussian random variables. The skeleton of a DAG is an undirected graphical model, which is constructed by removing the directions of all the edges in the DAG. Given observational data, not all the directions of the edges of a DAG are identifiable; however the skeleton of the DAG is identifiable. We propose a novel method named PenPC to estimate the skeleton of a high dimensional DAG by a two-step approach. We first estimate an undirected graph by selecting the non-zero entries of the partial correlation matrix, then remove false connections in this undirected graph to obtain the skeleton. We systematically study the asymptotic property of PenPC on high dimensional problems. Both simulations and real data analysis suggest that our method have substantially higher sensitivity and specificity to estimate network skeleton than existing methods.

To orient the edges in the skeleton of a DAG, we exploit interventional data on an additional set of variables. The variables are direct causes of some vertices in the DAG and enable estimating directions of the edges in the skeleton. More specifically, given the skeleton of a DAG, we calculate the posterior probabilities of edge directions using the additional set of variables. We evaluate our method by simulations and an application where variables modeled by a DAG are gene expression and the additional set variables are DNA polymorphisms.

Acknowledgments

I would like to express the deepest appreciation to my committee chair and advisor, Dr. Wei Sun, for his support and guidance. His constant encouragements and valuable advises have helped me though the entire process during my Ph.D. studies. He has been a great role model as a researcher and a teacher. Without his guidance and persistent help, this dissertation would not have been possible.

I would like to show my gratitude to Dr. Fred Wright, for his generous financial support and guidance on my first topic. His wide perspective on statistical genetics/genomics motivated me to strive for being a great researcher in the field. I received generous financial support from Dr. Joseph Ibrahim. I appreciate him for giving me an opportunity to be involved in an interesting project and valuable comments on my dissertation. My intellectual debt is to Dr. Michael Hudgens. His knowledge and insights in the field of causal inference have enriched my research. I owe my thanks to Dr. William Valdar for providing helpful suggestions in improving my dissertation.

I would like to give sincere thanks to Dr. Michael Kosorok, Dr. Jianwen Cai and Dr. Amy Herring for guiding me onto the right direction. Special thanks to all my friends with whom I have shared everything of our Ph.D. years.

Last but not the least, I am forever indebted to my father, Dr. Bok Dong Ha, my mother Duk Hee Mok and my sister Hyun Jin Ha for their unconditional love, support and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The relation of a set of random variables can be studied by graphical models, where the vertices represent the variables and edges capture the relations among the variables [Lauritzen, 1996]. A particular class of graphs, the directed acyclic graphs (DAGs) (also known as Bayesian Network) have been well studied for its importance in causal inference [Pearl, 2009]. In a DAG, all the edges are directed, and the direction of an edge implies a direct causal relation. There is no loop in a DAG, which is necessary to study causal relation [Spirtes et al., 2000]. Many methods have been developed to estimate DAGs from observational or interventional data, however, it remains a challenging problem in high dimensional setting where the number of variables can be larger or much larger than the sample size. The problem of DAG estimation in high-dimensional setting is the main focus of this dissertation.

Given observational data, a DAG is not identifiable, because conditional dependencies derived from observational data only determine the *skeleton* and *v-structures* of the graph [Pearl, 2009]. All the DAGs with the same skeleton and v-structures correspond to the same probability distribution and they form an equivalence class, which can be described by a completed partially directed acyclic graph (CPDAG) [Chickering, 2002]. Identification of v-structures after skeleton estimation only requires application of a set of deterministic rules. Therefore the tasks of estimating a CPDAG reduced

to estimating the skeleton of a DAG. Given a CPDAG, we can use the intervention calculus method developed by Maathuis et al. [2009] to infer causal effects.

We first focus on estimating partial correlation matrix which describes correlations between variables given all the remaining variables. Under multivariate Gaussian assumption, zero partial correlation implies independence of two variables given all the other variables. A partial correlation graph can be constructed by connecting variables with non-zero partial correlations. Suppose a DAG can model the relation of a set of variables which follow a multivariate Gaussian distribution. Then the partial correlation graph of these variables is closely related to the underlying DAG because the former can be considered as a *moral graph*, which is obtained by connecting two parents sharing a common child in the DAG and replacing all directed edges by undirected edges. Under the multivariate Gaussian assumption, we propose a method to estimate the skeleton of a DAG by estimating the corresponding sparse partial correlation matrix in the first step and applying a series of partial correlation testings in the second step. Finally, we consider to use external data of "surrogate experiments" to orient the DAG skeleton Bareinboim and Pearl [2012]. In such surrogate experiments, interventions are applied on an additional set of variables that are directed causes of the variables of interest.

The remaining part of the dissertation is organized as follows. Chapter 2 includes some definitions and notations for DAGs. In Chapter 3, we propose an estimation method of sparse partial correlation matrix using ridge regression followed by thresholding by hypothesis testing. In Chapter 4, we estimate the skeletons of high dimensional DAGs by a two-step algorithm called `PenPC`. Finally in Chapter 5, we develop a method to orient the edges in a DAG skeleton using interventional data in surrogate experiments.

Chapter 2

Preliminaries on Directed Acyclic Graphs (DAGs)

2.1 Assumptions

A directed graph denoted by $\mathcal{G}$ is a pair (V, E), where $V = \{1, \dots p\}$ is a finite set of vertices and E, the set of edges, is a subset of $(V \times V) \setminus \{(a, a)|a \in V\}$. The edge set E includes ordered pairs of distinct vertices and thus E includes no loops. A *path* of length n from i to j is a sequence $i = i_0 \to i_1 \to \cdots \to i_n = j$ of distinct vertices such that $(i_{l-1}, i_l) \in E$ for $l = 1, ..., n$. Given this path, i_{l-1} is a *parent* of i_l, i_l is a *child* of i_{l-1}, $i_0, i_1, ..., i_{l-1}$ are *ancestors* of i_l, and $i_{l+1}, ..., i_n$ are *descendants* of i_l. The graph $\mathcal{G}$ is called directed acyclic graph (DAG) if it contains no directed cyclic paths. Thus in a DAG, there is no path initiated from vertex i reaches i itself. The adjacency set of vertices of j, denoted by $\texttt{adj}(j, \mathcal{G})$, are the vertices that are connected to j by an edge of any directionality. A *chain* of length n from i to j is a sequence $i = i_0, i_1, \cdots, i_n = j$ of distinct vertices such that $i_{l-1} \to i_l$ or $i_l \to i_{l-1}$ for $l = 1, ..., n$.

Consider a DAG $\mathcal{G}$ whose vertices correspond to random variables $X_1, \dots, X_p$ and assume that

$$X = (X_1, \dots, X_p)^{\mathrm{T}} \in \mathbb{R}^p \sim P_X \text{ with density } f_X. \tag{2.1}$$

We say that the distribution P_X is *Markov* to $\mathcal{G}$ if the joint density f_X satisfies the

recursive factorization

$$f(x_1, \ldots, x_p) = \prod_{i=1}^{p} f(x_i | x_{\mathtt{pa}_i}), \tag{2.2}$$

where $\mathtt{pa}_i$ is defined by the set of parents of a vertex $i \in V$ in $\mathcal{G}$. The factorization naturally implies acyclic restriction of the graph structure. Equivalently P_X is Markov to $\mathcal{G}$ if every variable is conditionally independent of its non-descendants given its parents.

The *faithfulness* assumption requires stronger relationship between distribution P_X and DAG $\mathcal{G}$ than the Markov property.

Definition 1. *Let P_X be Markov to $\mathcal{G}$. $< \mathcal{G}, P_X >$ satisfies the faithfulness condtiontion if and only if every conditional independence relation true in P_X is entailed by the Markov property applied to $\mathcal{G}$ [Spirtes et al., 2000].*

This means that if a distribution P_X is faithful to DAG $\mathcal{G}$, all conditional independences can be read off from the DAG $\mathcal{G}$ using d-separation in the following definition 2.

Definition 2. *(d-separation). A vertex set $\mathbf{S}$ block a chain $\mathtt{p}$ if either (i) $\mathtt{p}$ contains at least one arrow-emitting vertex that is in $\mathbf{S}$, or (ii) $\mathtt{p}$ contains at least one collision vertex that is outside $\mathbf{S}$ and no descendant of the collision vertex belongs to $\mathbf{S}$. If $\mathbf{S}$ blocks all the chains from X to Y, it is said to "d-separate X and Y" [Pearl, 2009].*

The faithfulness assumption allows no extra conditional independence relations in the distribution P_X other than those which can be read from the DAG $\mathcal{G}$ using the d-separation. The more detailed description can be found in the literature [Robins et al., 2003]. Denote $\mathbf{P}_X(\mathcal{G})$ as all distributions that are Markov to $\mathcal{G}$. If $\mathcal{G}$ represents the data generating machanism for P_X, then P_X is Markov to G, in other words $P_X \in \mathbf{P}_X(\mathcal{G})$. Given $P_X \in \mathbf{P}_X(\mathcal{G})$, let $\mathbf{T}(P_X)$ represent all independence relationships for variable X under P_X. We say that P_X is faithful to $\mathcal{G}$ if $\mathbf{T}(P_X) = \cap_{Q \in \mathbf{P}_X(\mathcal{G})} \mathbf{T}(Q)$.

Not all the distributions can be faithfully represented by a DAG. If we assume that $X = (X_1, ..., X_p)^{\mathrm{T}} \in \mathbb{R}^p$ follow multivariate normal distribution, the non-faithful ones form a Lebesgue null set [Meek, 1995b] among all the multivariate normal distributions associated with $\mathcal{G}$.

2.2 Partial correlation graph as a moral graph of DAG

Consider an undirected graph $\mathcal{C} = (V, F)$, where $V = \{1, \dots p\}$ is a finite set of vertices corresponding to random variables $X = (X_1, \dots X_p)^T$ with an unknown covariance matrix $\mathbf{\Sigma}$ and F is a subset of $(V \times V) \setminus \{(a, a)|a \in V\}$ including unordered pairs of distinct vertices. The distribution P_X of X is said to *factorize* according to the undirected graph $\mathcal{C}$ if the joint density f_X satisfies

$$f_X(x_1, \dots, x_p) = \prod_{C \in \mathcal{C}} f_C(x_C), \tag{2.3}$$

where C is the set of cliques in $\mathcal{C}$ and f_C is the joint density of variables $X_C = \{X_i|i \in C \subseteq V\}$ [Lauritzen, 1996]. The undirected graph which satisfies the factorization property has Markov property: for any pair of vertices (i, j), $(i, j) \in F$ if and only if X_i and X_j are conditionally independent given all the remaining variables $\{X_k|k \in V \setminus \{i, j\}\}$. The graph satisfying this Markov property is called *independence graph*. The partial correlation graph under the Gaussian assumption is independence graph and is often called Gaussian graphical model (GGM). The covariance selection problem Dempster [1972] is equivalent to the estimation of $\mathcal{C}$ because conditional independence relations implied by the factorization on $\mathcal{C}$ can be identified by the zero structure of the inverse covariance matrix denoted by $\mathbf{\Omega}$ under the normality assumption.

For a DAG $\mathcal{G} = (V, E)$ we can define its moral graph for the same set of vertices V as an undirected graph constructed by connecting parents with a common child

and subsequently deleting directions on all edges [Lauritzen, 1996]. Assuming the factorizations both on $\mathcal{G}$ and $\mathcal{C}$, it is easily seen that the moral graph of $\mathcal{G}$ is $\mathcal{C}$ by lemma 3.21 of [Lauritzen, 1996].

2.3 Identifiability of DAG

A DAG $\mathcal{G}$ is not identifiable from the distribution P_X which is assumed to be Markov to $\mathcal{G}$ because several different DAG's may determine the same P_X. In other words, because several different DAGs may determine the same set of conditional independence restrictions among a given set of random variables, the collection of all possible DAGs for these variables naturally coalesces into one or more classes of *Markov equivalent* DAGs, where all DAGs within a Markov class determine the same statistical model (the same factorization) [Andersson et al., 1997]. The following theorem well characterizes the Markov equivalence class.

Theorem 1. *Two DAG's are Markov equivalent if and only if they have the same skeleton and the same v-structure [Andersson et al., 1997].*

The *skeleton* of a DAG $\mathcal{G}$ is obtained by replacing all directed edges to undirected edges: the skeleton is denoted by $\mathcal{G}^u = (V, E^u)$ where $(i,j) \in E^u \Leftrightarrow (i,j) \in E$ or $(j,i) \in E$. The *v-structure* is an ordered triple of vertices (i,j,k) such that $\mathcal{G}$ contains the directed edges $(i,k) \in E$ and $(j,k) \in E$ and i and j are not adjacent in $\mathcal{G}$: in this v-structure, the co-parents i and j share a common child k which is called a *collision* vertex. The distribution P_X is faithful to $\mathcal{G}$ if and only if (i) for any vertex pair (i, j) in V, $(i,j) \in E^u$ if and only if i and j are dependent conditional on every subset in $V \setminus \{i,j\}$ and (ii) in a *v-structure* $i \rightarrow k \leftarrow j$, i and j are marginally independent or conditionally independent given the parents of i and j, but i and j are dependent with each other given every set that contain k (a collision vertex) or its descendants but not i or j.

A partially directed acyclic graph (PDAG) is a graph that contains both directed and undirected edges and no directed cycle. The Markov equivalence class corresponding to a DAG can be represented by a PDAG [Chickering, 2002]. Specifically, from Theorem 1, a PDAG representing a Markov equivalence class is constructed by replacing the undirected edges of the skeleton with directed edges for every edge participating v-structures. Then the undirected edges of the PDAG can be maximally oriented while keeping the v-structures and maintaining acyclic constraints. It is called completed PDAG (CPDAG). The CPDAG corresponding to an equivalence class is the PDAG consisting of directed edges which exist in every DAGs belonging to the equivalence class and undirected edges for reversible edges in the equivalence class.

Chapter 3

Partial Correlation Matrix Estimation Using Ridge Penalty Followed by Hypothesis Testing

3.1 Introduction

The expression of multiple genes can be studied through a network perspective, where the set of genes of interest are vertices and the relations among the genes are undirected/directed edges. The gene coexpression network analysis is a popular approach to dissect gene expression regulation patterns and to detect functionally related genes [Stuart et al., 2003; de Jong et al., 2012]. In this chapter we study the (undirected) co-expression network of a group of genes constructed through their partial correlation matrix, where the partial correlation of two genes is a measure of the linear relationship of these two genes' expression conditioning on the expression of all the other genes.

We consider a p-dimensional random vector (i.e., the expression of p genes) $X = (X_1, ..., X_p)^{\mathrm{T}} \in \mathbb{R}^p$ with an unknown covariance matrix $\mathbf{\Sigma}$. Assuming that the covariance matrix $\mathbf{\Sigma}$ is positive definite, let $\mathbf{\Omega} = [\mathbf{\Omega}_{ab}]_{p\times p} = \mathbf{\Sigma}^{-1}$ be the inverse of the covariance matrix. $\mathbf{\Omega}$ is also called concentration matrix or precision matrix. The partial correlations can be obtained by the off diagonal elements of the negative definite matrix

$$\mathbf{R} = [\rho_{ab}]_{p\times p} = -\texttt{scale}(\mathbf{\Omega}), \tag{3.1}$$

where the `scale` is an operator defined for a square matrix. Let $\texttt{diag}(\mathbf{A})$ be a diagonal matrix constructed by the diagonal elements of $\mathbf{A}$, then

$$\texttt{scale}(\mathbf{A}) = \texttt{diag}(\mathbf{A})^{-1/2}\mathbf{A}\texttt{diag}(\mathbf{A})^{-1/2}.$$

The derivation of equation (3.1) is presented in the Appendix I. The zero structure of the partial correlation matrix of p random variables can be represented by an undirected graph

$$\mathbf{G} = (\mathbf{\Gamma}, \mathbf{E}),$$

where $\mathbf{\Gamma} = \{1, \cdots, p\}$ is the set of vertices and $\mathbf{E}$ is a set of edges in $\mathbf{\Gamma} \times \mathbf{\Gamma}$ such that any edge between vertices a and b belongs to $\mathbf{E}$ if and only if $\rho_{ab} \neq 0$, i.e, the two random variables X_a and X_b are conditionally correlated given all the remaining variables $X_{\mathbf{\Gamma}\setminus\{a,b\}} = \{X_k : k \in \mathbf{\Gamma} \setminus \{a, b\}\}$. We refer to such an undirected graph $\mathbf{G}$ as a *partial correlation graph*. Under multivariate Gaussian distribution assumption for X, the zero off-diagonal entries of $\mathbf{\Omega}$ or $\mathbf{R}$ occur if and only if the corresponding variables are conditionally independent given the remaining variables.

Although many methods have been developed for partial correlation matrix estimation in high dimensional problems where $p > n$, we find that a simple penalized estimation using ridge penalty has favorable error properties. The advantage of this ridge penalization approach has not been appreciated in the existing literature, partly because it does not provide sparse estimates, i.e., none of the partial correlation is estimated exactly as 0. We propose a novel approach to threshold the ridge estimates to decide which partial correlation estimates are not 0's by hypothesis testing. The null distribution of our test statistics is estimated from the observed test statistics to provide appropriate control of type I error. Finally we re-estimate the partial correlation coefficients on the none-zero entries of the partial correlation matrix. Thresholding

ridge estimates is desirable because it leads to parsimonious and more interpretable partial correlation matrix estimate. Such thresholding often further reduces estimation error, and provides a logical connection between estimation and testing.

Next we briefly review the existing works for estimating concentration matrix or partial correlation matrix and related statistical inference. Assume $X = (X_1, ..., X_p)^{\mathrm{T}}$ follows multivariate Gaussian distribution, denoted by $N(\mu, \mathbf{\Sigma})$. Suppose there are n independent samples of X. Let $\mathbf{X}$ be the $p \times n$ centered data matrix. Assuming $\mu = \mathbf{0}$, the log-likelihood of concentration matrix $\mathbf{\Omega}$ is proportional to

$$l(\mathbf{\Omega}) = \log |\mathbf{\Omega}| - \mathtt{tr}(\mathbf{S\Omega}), \tag{3.2}$$

where $\mathtt{tr}(\cdot)$ is trace of a square matrix and $\mathbf{S} = \mathbf{X}\mathbf{X}^{\mathrm{T}}/n$ is the sample covariance matrix. When $n \geq p$, $\mathbf{S}$ is positive definite with probability 1 and $\mathbf{S}^{-1}$ is the maximum likelihood estimate (MLE) of $\mathbf{\Omega}$ [Lauritzen, 1996]. However, this approach fails when $p > n$, and may perform poorly unless n is much larger than p. Therefore MLE with certain constraints or penalized MLE are often used for high dimensional problems when p is lager or much larger than n. Examples include covariance selection from positive definite matrices [Dempster, 1972] or iterative partial maximization based on deviance tests [Speed and Kiiveri, 1986]. More general linear restrictions on edges are enabled by colored graph models [Højsgaard and Lauritzen, 2008]. Recently, many penalized MLE of $\mathbf{\Omega}$ have been proposed for high dimensional problems [Yuan and Lin, 2007; Rothman et al., 2008; Banerjee et al., 2008; Friedman et al., 2008; Fan et al., 2009]. One of the most widely used methods is the graphic Lasso [Friedman et al., 2008], which maximizes the following penalized log likelihood:

$$l(\mathbf{\Omega}) = \log |\mathbf{\Omega}| - \mathtt{tr}(\mathbf{S\Omega}) - \kappa \sum_{a,b} |\mathbf{\Omega}_{ab}|, \tag{3.3}$$

where κ is a tuning parameter.

With a focus on determining the partial correlation graph, rather than precise estimation of the partial correlation coefficients, Meinshausen and Bühlmann [2006] proposed a regression-based approach called *neighborhood selection*. The neighborhood for each vertex was estimated by penalized regression of the corresponding variable versus the remaining variables. Banerjee et al. [2008]; Friedman et al. [2008] showed that estimating the penalized MLE (with L_1 penalty) of $\mathbf{\Omega}$ could be viewed as p-coupled iterative versions of the p separate neighborhood selections. More recent methodology developments related with neighborhood selection include Yuan [2010] and Zhou et al. [2011].

Statistical inference of partial correlation estimates is another topic related with our method development. Given a partial correlation estimate, denoted by $\hat{\rho}$, one may test $H_0 : \rho = 0$ against $H_A : \rho \neq 0$ using a test statistic constructed by Fisher's Z-transformation: $\psi(\hat{\rho}) = 0.5 \log \{(1 + \hat{\rho}) / (1 - \hat{\rho})\}$. Specifically, one may reject the null hypothesis at level α if $(n - p - 1)^{1/2} |\psi(\hat{\rho})| > \Phi^{-1}(1 - \alpha/2)$ for standard normal c.d.f. Φ [Anderson, 2003]. However, this testing procedure assumes the sample size n is substantially greater than p. For high dimensional problems with $p > n$, Schäfer and Strimmer [2005] proposed to estimate partial correlation matrix using a combination of Bootstrap aggregation and pseudoinverse. Then they made inference by assuming their partial correlation estimates across all variables followed a mixture of null and alternative distributions where the null was classical asymptotic distribution of partial correlation [Hotelling, 1953] with unknown degree of freedom and the alternative was uniform (-1,1). Magwene et al. [2004] and Wille et al. [2004] proposed to use low-order partial correlations to avoid singularity problem when $p > n$. Subsequently, Wille and Bühlmann [2006] discussed more formal statements on Gaussian graphical model inference using low-order partial correlations. Castelo and Roverato [2006] generalized

the 0-1 partial correlation graph to an arbitrary $q \leq p-2$ order partial correlation graph.

The remaining parts of the chapter are organized as follows. We present our method in Section 3.2, demonstrate the effectiveness of our method by simulations and real data analysis in Section 3.3, and conclude this chapter by some discussions in Section 3.4.

3.2 Method

3.2.1 Estimation of partial correlation matrix using ridge penalty

We assume the $p \times n$ data matrix $\mathbf{X}$ has been centered so that $\mathbf{S} = \mathbf{X}\mathbf{X}^{\mathrm{T}}/n$ is the sample covariance matrix. Then a straightforward estimate of (the off-diagonal elements of) a partial correlation matrix can be obtained from

$$\hat{\mathbf{R}} = -\mathtt{scale}(\mathbf{S}^{-1}).$$

However, when $n < p$, $\mathbf{S}$ is not invertible. To solve the singularity problem of inverting a sample covariance matrix, we add a positive constant to the diagonal elements of the sample covariance matrix:

$$\hat{\mathbf{R}}(\lambda) = -\mathtt{scale}\big((\mathbf{S} + \lambda \mathbf{I}_p)^{-1}\big), \tag{3.4}$$

where $\lambda \geq 0$ and $\mathbf{I}_p$ is a $p \times p$ identity matrix. We call $\mathbf{S}^{+}(\lambda) = (\mathbf{S} + \lambda \mathbf{I}_p)^{-1}$ as the *ridge* inverse in the analogy to ridge regression [Hoerl and Kennard, 1970]. The modified sample covariance matrix $\mathbf{S} + \lambda \mathbf{I}_p$ guarantees full rank for any $\lambda > 0$, and has been used as an initial covariance matrix estimate in the coordinate descent algorithms in Banerjee et al. [2008]; Friedman et al. [2008].

Next we show that as λ varies from 0 to ∞, $\hat{\mathbf{R}}(\lambda)$ varies from a scaled generalized

inverse to an identity matrix. Let $\mathbf{X}/\sqrt{n} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$ be a singular value decomposition with $\mathtt{rank}(\mathbf{X}) = k \leq \mathtt{min}(n, p)$, where $\mathbf{U}$ and $\mathbf{V}$ are, respectively $p \times p$ and $n \times n$ orthogonal matrices, $\mathbf{D}$ is $p \times n$ diagonal matrix with its first k nonzero diagonal elements $d_1, \ldots, d_k$ and all other elements being zero. Since $\mathbf{S}^{+}(\lambda) = \mathbf{U}(\mathbf{D} + \lambda \mathbf{I}_p)^{-1}\mathbf{U}^{\mathrm{T}}$, it is obvious that

$$\mathtt{scale}\left(\mathbf{S}^{+}(\lambda)\right) \to \mathtt{scale}(\mathbf{S}^{-}) \text{ as } \lambda \to 0, \tag{3.5}$$

where $\mathbf{S}^{-}$ is Moore-Penrose generalized (MPG) inverse of $\mathbf{S}$ if $k < p$ [Schott, 2005]. By the invariance of the `scale` operator under scalar product,

$$\mathtt{scale}\left(\mathbf{S}^{+}(\lambda)\right) = \mathtt{scale}\left(\lambda \mathbf{S}^{+}(\lambda)\right) \to \mathbf{I}_p \text{ as } \lambda \to \infty. \tag{3.6}$$

Since the estimates of regression coefficients using MPG inverse is minimum L_2 solution (proposition 1 of Lv and Fan [2009]), $\mathbf{S}^{+}(\lambda)$ goes to k rank ridge inverse when λ goes to 0 by (3.5). From (3.6) the partial correlation matrix shrinks toward the identity matrix as λ goes to infinity.

$\mathbf{S}^{+}(\lambda)$ can also be understood as the inverse of a shrinkage estimate of the covariance matrix [Schäfer et al., 2005] since we can rewrite $\mathbf{S}^{+}(\lambda)$ as

$$c\mathbf{S}^{+}(\lambda) = \left((1 - \lambda')\mathbf{S} + \lambda' \mathbf{I}_p\right)^{-1}, \tag{3.7}$$

where $\lambda' = \lambda/(1 + \lambda)$ and $c = (1 + \lambda)$.

We choose the tuning parameter λ in the ridge inverse by its equivalence relationship with p separate ridge regressions. Given λ, for any $a \in \mathbf{\Gamma}$, the ridge coefficients are

$$\hat{\boldsymbol{\beta}}^{a}(\lambda) = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p-1}}{\arg\min} \frac{1}{n}(\mathbf{X}_a - \mathbf{X}_{-a}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{X}_a - \mathbf{X}_{-a}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta},$$

where $\mathbf{X}_a$ is an $n \times 1$ vector for n measurements of variable X_a, $\mathbf{X}_{-a}$ is an $n \times (p-1)$

matrix for n measurements of the remaining $p-1$ variables X_{-a}. Given the following decomposition of the sample covariance matrix $\mathbf{S}$ with respect to variables (X_a, X_{-a}),

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_{a,a} & \mathbf{S}_{a,-a} \\ \mathbf{S}_{-a,a} & \mathbf{S}_{-a,-a} \end{pmatrix},$$

$\hat{\boldsymbol{\beta}}^a(\lambda)$ has a closed-form solution

$$\hat{\boldsymbol{\beta}}^a(\lambda) = (\mathbf{S}_{-a,-a} + \lambda \mathbf{I}_{p-1})^{-1} \mathbf{S}_{-a,a}.$$

Now we show that estimating the off-diagonal elements of $\hat{\mathbf{R}}(\lambda)$ is the same as estimating regression coefficients using p separate ridge regressions. Let

$$\mathbf{W} = \mathbf{S}^+(\lambda) = \begin{pmatrix} \mathbf{S}_{a,a} + \lambda & \mathbf{S}_{a,-a} \\ \mathbf{S}_{-a,a} & \mathbf{S}_{-a,-a} + \lambda \mathbf{I}_{p-1} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{W}_{a,a} & \mathbf{W}_{a,-a} \\ \mathbf{W}_{-a,a} & \mathbf{W}_{-a,-a} \end{pmatrix}.$$

From the inverse formula for block matrices,

$$\mathbf{W}_{-a,a} = -\mathbf{W}_{a,a}(\mathbf{S}_{-a,-a} + \lambda \mathbf{I}_{p-1})^{-1} \mathbf{S}_{-a,a} = -\mathbf{W}_{a,a} \hat{\boldsymbol{\beta}}^a(\lambda).$$

Therefore the estimation of partial correlation matrix is equivalent to estimating the regression coefficients p separate ridge regressions. To choose the tuning parameter λ, we minimize 10-fold cross validation estimates of the total prediction errors of the p ridge regressions.

3.2.2 Thresholding

We propose a hypothesis testing approach to threshold the off-diagonal elements of the ridge estimates $\hat{\mathbf{R}}(\lambda) = [\hat{\rho}^{\lambda}_{ab}]_{p \times p}$. We use Fisher's Z-transformation of partial

correlations as our test statistics, denoted by $\{\psi(\hat{\rho}^{\lambda}_{ab}) : a \in \mathbf{\Gamma}, b \in \mathbf{\Gamma}, \text{ and } a \neq b\}$. By taking advantage of the sparsity of the high dimensional partial correlation matrix, we estimate the null distribution of our test statistics from data using Efron's central matching method. Although our testing approach can apply to a variety of partial correlation estimates, we develop it here for the ridge inverse estimate.

Following Efron [2004], we assume the observed test statistics follow a *mixture* distribution

$$f(\psi) = (1-\eta)f_0(\psi) + \eta f_a(\psi), \tag{3.8}$$

where the null distribution $f_0(\psi)$ is a normal distribution $N(\mu_0, \sigma_0^2)$, the alternative distribution $f_a(\psi)$ is left un-specified, and η is the proportion of the observations arising from the non-null distribution. We also assume that a large proportion of the observed ψ values are from the null distribution, i.e. $\eta \approx 0$. This assumption reflects the belief that the partial correlation matrix is sparse. Then the central part of the marginal distribution of ψ is mostly occupied by the observations from the null distribution and only the tail areas of the marginal distribution are affected by the small proportion of non-null observations. Therefore using Efron's central matching method, we can estimate the null distribution (i.e., estimate μ_0 and σ_0) by matching the marginal distribution and the null distribution at the center part of the distributions. Specifically, assuming $f(\psi) = f_0(\psi)$ around $\psi = 0$ gives

$$\log f(\psi) = -\frac{1}{2}\Big(\frac{\psi - \mu_0}{\sigma_0}\Big)^2 + C \tag{3.9}$$

for a constant C.

We estimate the density $f(\psi)$ using polynomial poisson regression. The range of the $p(p-1)/2$ observed ψ values is partitioned into K equal intervals with interval k having mid point x_k and s_k observed ψ values. s_k's (k=1,...,K) are assumed to be

independently distributed following Poisson distributions with mean ν_k's. We fit a q degree polynomial Poisson regression on ν_k,

$$\log(\nu_k) = \log\big(f(x_k)/c\big) = \sum_{j=1}^{q} \theta_j (x_k)^j, \tag{3.10}$$

for $k = 1, ..., K$ and a normalizing constant c making the marginal density $f(\psi)$ integrated to 1. The estimates of $\{\theta_j : j = 1, \ldots, q\}$ are used to estimate $\log\big(\hat{f}(\psi)/c\big) = \sum_{j=1}^{q} \hat{\theta}_j \psi^j$. Then using equation (3.9), we can obtain the estimates of μ_0 and σ_0:

$$\hat{\mu}_0 = \arg\max\Big\{\hat{f}(\psi)\Big\}, \quad \hat{\sigma}_0 = \left[-\frac{d^2}{d\psi^2}\log\hat{f}(\psi)\right]^{-\frac{1}{2}}_{\psi=\hat{\mu}_0}. \tag{3.11}$$

The degree of polynomial regression, q, is a nuisance parameter. Based on the sparsity assumption that most p-values arise from the null, we choose the q so that the p-values are most uniformly distributed. The empirical distribution function of the p-values, $\{\pi_{ab}^{(q)} | a \neq b \in \mathbf{\Gamma}\}$ given q, is

$$F_q(\pi) = \frac{2}{p(p-1)} \sum_{a,b \in \mathbf{\Gamma}, a \neq b} I(\pi_{ab}^{(q)} \leq \pi). \tag{3.12}$$

We suggest to estimate q by

$$\hat{q} = \arg\min_{q} \Big[\sup_{0<\pi<1} |F_q(\pi) - F_0(\pi)| \Big], \tag{3.13}$$

where $F_0(\pi)$ is uniform distribution between 0 and 1, and $D_q = \sup_{0<\pi<1} |F_q(\pi) - F_0(\pi)|$ is a distance measure used in Kolmogorov-Smirnov statistic. Figure 3.1 displays the average and one standard deviation of D_q values over 100 simulation data sets for $p = 500, n = 30$ and $\eta = 0$ or $\eta = 0.0003$, which corresponds to 38 non-zero partial correlations. Adding 38 nonzero partial correlations to the null needs 3 or 4 higher

polynomial order on average to estimate the null distribution.

3.2.3 Re-estimation of partial correlation coefficients

Given the zero structure estimated in the previous step, we re-estimate the partial correlation coefficients at the non-zero entries of the partial correlation matrix. Suppose that the covariance matrix $\mathbf{\Sigma}$ and the concentration matrix $\mathbf{\Omega}$ are partitioned according to random variables (X_a, X_{-a}) and the blocks are denoted by $\mathbf{\Sigma}_{a,a}$, $\mathbf{\Sigma}_{-a,a}$, $\mathbf{\Sigma}_{a,-a}$, $\mathbf{\Sigma}_{-a,-a}$ and $\mathbf{\Omega}_{a,a}$, $\mathbf{\Omega}_{-a,a}$, $\mathbf{\Omega}_{a,-a}$, $\mathbf{\Omega}_{-a,-a}$. Consider the best linear predictor of X_a by $X_{-a}^{\mathrm{T}}\boldsymbol{\beta}^a$ for any $a \in \mathbf{\Gamma}$. Let $\epsilon_a = X_a - X_{-a}^{\mathrm{T}}\boldsymbol{\beta}^a$. It is easy to show that $\boldsymbol{\beta}^a = \mathbf{\Sigma}_{-a,-a}^{-1}\mathbf{\Sigma}_{-a,a}$ and $\mathtt{Var}(X_a - X_{-a}^{\mathrm{T}}\boldsymbol{\beta}^a) = \mathtt{Var}(\epsilon_a) = \mathbf{\Sigma}_{a,a} - \mathbf{\Sigma}_{a,-a}\mathbf{\Sigma}_{-a,-a}^{-1}\mathbf{\Sigma}_{-a,a}$. From inverse formula for block matrix and $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$,

$$\mathbf{\Omega}_{a,a} = (\mathtt{Var}(\epsilon_a))^{-1}, \quad \mathbf{\Omega}_{-a,a} = -(\mathtt{Var}(\epsilon_a))^{-1}\boldsymbol{\beta}^a. \tag{3.14}$$

From $\hat{\mathbf{E}}(\lambda, \alpha)$ estimated in the thresholding step, we know all the variables adjacent to $a \in \mathbf{\Gamma}$, denoted by $\hat{\mathtt{ne}}_a$. Based on the sparsity assumption, we assume $|\hat{\mathtt{ne}}_a| < n$, then we can have the following refined estimates of the concentration matrix:

$$\tilde{\mathbf{\Omega}}_{aa} = (n - |\hat{\mathtt{ne}}_a|)/\|\mathbf{X}_a - \mathbf{X}_{\hat{\mathtt{ne}}_a}\hat{\boldsymbol{\beta}}^{a,\hat{\mathtt{ne}}_a}\|_2^2, \quad \tilde{\mathbf{\Omega}}_{-aa} = -\tilde{\mathbf{\Omega}}_{aa}\hat{\boldsymbol{\beta}}^{a,\hat{\mathtt{ne}}_a}, \tag{3.15}$$

where $\hat{\boldsymbol{\beta}}^{a,\hat{\mathtt{ne}}_a} = \left(\mathbf{X}_{\hat{\mathtt{ne}}_a}\mathbf{X}_{\hat{\mathtt{ne}}_a}^{\mathrm{T}}\right)^{-}\mathbf{X}_{\hat{\mathtt{ne}}_a}\mathbf{X}_a$, $\mathbf{X}_{\hat{\mathtt{ne}}_a}$ is $|\hat{\mathtt{ne}}_a| \times n$ submatrix of $\mathbf{X}$ corresponding to $\hat{\mathtt{ne}}_a$, and $(\cdot)^{-}$ is the $k = \min(n, |\hat{\mathtt{ne}}_a|)$ rank ridge inverse using MPG inverse. Since this solution is not symmetric in general, we set the final estimates of the off-diagonal elements of $\mathbf{\Omega}$ as

$$\hat{\mathbf{\Omega}}_{a,b} = \hat{\mathbf{\Omega}}_{b,a} = \mathtt{sign}(\hat{\boldsymbol{\beta}}_b^{a,\hat{\mathtt{ne}}_a})\sqrt{\tilde{\mathbf{\Omega}}_{a,b}\tilde{\mathbf{\Omega}}_{b,a}} \text{ for } a \neq b, \tag{3.16}$$

and the partial correlation coefficients estimates from $-\texttt{scale}(\hat{\boldsymbol{\Omega}})$.

3.3 Results

3.3.1 Simulation I

We first use a simple simulation to demonstrate that the p-values calculated using central matching method follow the expected uniform distribution under null, while the p-values calculated using asymptotic distribution can lead to inflated type I error. We simulated data from multivariate Gaussian distribution $N(\mathbf{0}, \boldsymbol{I}_{p\times p})$ with $p = 100$ and $n = 1000$ or 110. All pairwise partial correlations were calculated by inverting the sample correlation matrix, and then the test statistics were calculated by Fisher's Z transformation of the partial correlations. The p-values of the test statistics were calculated using theoretical null distribution $N(0, 1/n - p - 1)$, and the empirical null distribution estimated by central matching method. As shown in the qq-plots of Figure 3.2, when $p = 100$ and $n = 1000$, p-values calculated using either the theoretical null distribution or the empirical null distribution followed the expected uniform distribution. However when the sample size was decreased to $n = 110$, the p-values calculated from the empirical null distribution were still uniformly distributed but the p-values calculated from the theoretical null distribution were severely inflated.

3.3.2 Simulation II

We consider random networks where both the network structure and the partial correlation coefficients are random. The only restriction is that the partial correlation matrix is diagonally dominant, so that $\mathbf{R}$ is a strictly negative definite matrix. The simulation datasets were generated following similar approach of Schäfer and Strimmer [2005]. We simulated a $p\times n$ data matrix $\mathbf{X}$ composed of n independent random samples from p dimensional multivariate Gaussian distribution $N_p(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is determined

by a simulated concentration matrix $\mathbf{\Omega}$. We initialized $\mathbf{\Omega}$ by a $p \times p$ matrix with all elements being 0's. Given η, the proportion of non-null edges among all the $p(p-1)/2$ edges, we randomly selected $100\eta\%$ of the off-diagonal elements of $\mathbf{\Omega}$ and filled in values from uniform distribution on [-1,1]. To ensure that $\mathbf{\Omega}$ is a positive definite matrix, the diagonal elements of $\mathbf{\Omega}$ were filled by column-wise sums of absolute values plus a small constant. Finally $\mathbf{\Sigma}$ was calculated by $\texttt{scale}(\mathbf{\Omega}^{-1})$.

Let $|\mathbf{E}|$ be the number of edges in set $\mathbf{E}$. Our simulation settings are

1. $p = 50$, $n = 100$, and $|\mathbf{E}| = 45, 55, 65, 75$

2. $p = 200$, $n = 100$, and $|\mathbf{E}| = 160, 200, 220, 240$.

We evaluated the accuracy of partial correlation graph using ROC curve, and the accuracy of partial correlation coefficient estimates using sum squared error (SSE). Given the set of vertices $\mathbf{\Gamma} = \{1, 2, ..., p\}$, The SSE was calculated as

$$L(\mathbf{R}, \hat{\mathbf{R}}) = \sum_{a \neq b \in \mathbf{\Gamma}} (\hat{\rho}_{ab} - \rho_{ab})^2, \tag{3.17}$$

where $\hat{\mathbf{R}} = [\hat{\rho}_{ab}]_{p \times p}$ was the estimates of $\mathbf{R}$. For either the ROC curve of the SSE value, the mean values calculated from 100 replicates of each simulation setting were reported.

We compared our method with one of the most widely used method for partial correlation matrix estimation, the Graphical Lasso (Glasso) [Friedman et al., 2008]. For our method, we used 10-fold cross-validation to choose the value of ridge parameter λ. After estimating the edge set denoted by $\hat{\mathbf{E}}(\lambda, \alpha)$ using ridge parameter λ and the p-value thresholding level α, we re-estimated the nonzero partial correlations under the sparsity implied by $\hat{\mathbf{E}}(\lambda, \alpha)$. The results of all simulation settings are displayed in Figure 3.3-Figure 3.10. For example, Figure 3.3 and Figure 3.7 show the ROC curves in panel (a) and the SSE curves in panel (b) across various threshold values α for our

method and across different values of the tuning parameter κ of the Glasso. In the ROC curves for both high and low dimension cases, our method has uniformly better sensitivity and and specificity than the Glasso in estimating the network structure. The SSE curves show that our method attains lower SSE value than the Glasso around the true sparsity level.

3.3.3 Application

We applied our method to estimate the partial correlation graph of the expression of 6178 genes from yeast cell cycle data [Spellman et al., 1998]. The gene expression data were downloaded from `http://genome-www.stanford.edu/cellcycle/data/rawdata/`. After removing the samples with more than 20% missing values, 75 samples remained for further analysis. We imputed the remaining missing values of the expression data using nearest neighbor averaging. Then the expression data of each sample were normalized by quantile normalization. In this analysis we did not account for the time-dependent nature of the data and treated the expression of each gene across 75 samples as independent observations.

Denote the observed gene expression data as a matrix $\mathbf{X}$ of dimension 6178×75. Each gene is a variable, and thus $\mathbf{\Gamma}$ is $\{1, ..., 6178\}$. We first grouped the 6178 genes into h clusters. Let C_i be the genes belonging to the i-th cluster, then $\sum_{i=1}^{h} |C_i| = 6178$. We separately constructed the partial correlation graph within each cluster. The graph of ith cluster is denoted by $G_{C_i} = (C_i, E_{C_i})$ for $i = 1, \ldots, h$. We assumed the genes from different clusters were independent, so that the edge set $\mathbf{E}$ of the whole graph was estimated by $\hat{\mathbf{E}} = \bigcup_{i=1}^{h} \hat{E}_{C_i}$. Specifically, we clustered the 6178 genes using hierarchical clustering with Ward's minimum variance method and the distance between two genes a and b was defined as $1 - |\hat{\rho}_{ab|\emptyset}|$ where $\hat{\rho}_{ab|\emptyset}$ denoted marginal Pearson correlation. We chose the number of clusters to be 25 based on the *gap* statistic [Tibshirani et al., 2001].

Cluster sizes varied from 32 to 1370, with 25 percentile, median, and 75 percentile being 139, 194, and 254, respectively. Among all the 19,080,753 gene pairs of the 6178 genes, 1,556,154 belonged to the same cluster, and hence could be connected based on the partial correlation graph estimates.

We compared the performance of our method with the Glasso [Friedman et al., 2008]. We chose the tuning parameter κ of Glasso based on the extended Bayesian information criterion (BIC) [Foygel and Drton, 2010]:

$$\texttt{BIC}_\gamma(\kappa) = -\log|\hat{\mathbf{\Omega}}(\kappa)| + \texttt{tr}(\hat{\mathbf{\Omega}}(\kappa)\mathbf{S}) + \frac{1}{n}|\hat{\mathbf{E}}(\kappa)|\log n + \frac{4}{n}|\hat{\mathbf{E}}(\kappa)|\gamma\log p, \tag{3.18}$$

where $\hat{\mathbf{\Omega}}(\kappa)$ was the estimate of the inverse covariance matrix using Glasso with tuning parameter κ, $\hat{\mathbf{E}}(\kappa)$ was the edge set obtained from $\hat{\mathbf{\Omega}}(\kappa)$, and $\gamma \in [0, 1]$ is a tuning parameter of the extended BIC. If $\gamma = 0$, the classical BIC used in Yuan and Lin [2007] was recovered. Given a fixed γ value, we applied Glasso with tuning parameter selected by the extended BIC (3.18) to construct the partial correlation graphs for all 25 clusters. An exception is that for two clusters with low dimension such that $p < n/2$, we always used the classical BIC by setting $\gamma = 0$. Different choices of γ led to the different model selection results.

The estimates of partial correlation graphs were evaluated by comparing the edge set $\hat{\mathbf{E}}$ with yeast protein-protein interaction database at `http://thebiogrid.org/download.php`. Table 3.1 displays the number of directed edges, the number of undirected edges (after omitting the directions), and the number of vertices in each of 20 protein-protein interaction dataset. We considered two genes connected if they belonged to the same cluster and the corresponding proteins had interaction according to at least one of the protein-protein interaction datasets. Among 1,556,154 gene pairs belonging to the same cluster, 9,382 were connected and 1,546,772 were not connected. Given this imperfect, but biologically meaningful definition of true/false connections, we evaluated

our method and Glasso by ROC curves. The ROC curve of our method was generated across different p-value cutoffs α and the ROC curve of the Glasso was generated across different γ values in the extended BIC. Our method had uniformly higher sensitivity and specificity than the Glasso (Figure 3.11) to predict protein-protein interactions in the database.

3.4 Discussion

We have described a new framework for estimation and statistical inference of partial correlation matrix. Both simulation and real data analysis have demonstrated the effectiveness of our method. For real data analysis where p is much larger than n, we cluster the genes and then estimate partial correlation matrix within each cluster. This is based on an reasonable assumption that the partial correlation matrix of gene expression has a block diagonal structure. We used the hierarchical clustering to group genes. There are many other clustering method available Monti et al. [2003]; Zhang et al. [2005], though a careful study of which clustering method can better identify the block diagonal structure is beyond the scope of this chapter. Our method does not require multivariate Gaussian distribution assumption. However, without this assumption, partial correlation being zero may not imply the two variables are independent with each other.

3.5 Tables and figures

Table 3.1: Summary of the protein-protein interaction database

ID	Experiment system (type)	no. of directed edges	no. of undirected edges	no. of vertices
1	Affinity Capture-MS (physical)	72767	42538	4613
2	Affinity Capture-Western (physical)	13105	7795	2727
3	Dosage Rescue (genetic)	4812	4022	2161
4	Reconstituted Complex (physical)	5110	3946	1988
5	Synthetic Lethality (genetic)	13870	10965	2915
6	Two-hybrid (physical)	13986	10827	3392
7	Biochemical Activity (physical)	5703	5220	1946
8	Co-crystal Structure (physical)	387	337	421
9	FRET (physical)	142	119	117
10	Protein-peptide (physical)	673	643	353
11	Co-localization (physical)	527	484	441
12	Affinity Capture-RNA (physical)	5895	5888	3702
13	Protein-RNA (physical)	408	399	377
14	PCA (physical)	5117	4845	1663
15	Co-purification (physical)	1675	1309	933
16	Co-fractionation (physical)	777	725	663
17	Dosage Lethality (genetic)	971	945	786
18	Phenotypic Enhancement (genetic)	6449	4803	2153
19	Phenotypic Suppression (genetic)	5287	3965	1729
20	Synthetic Haploinsufficiency (genetic)	262	262	262

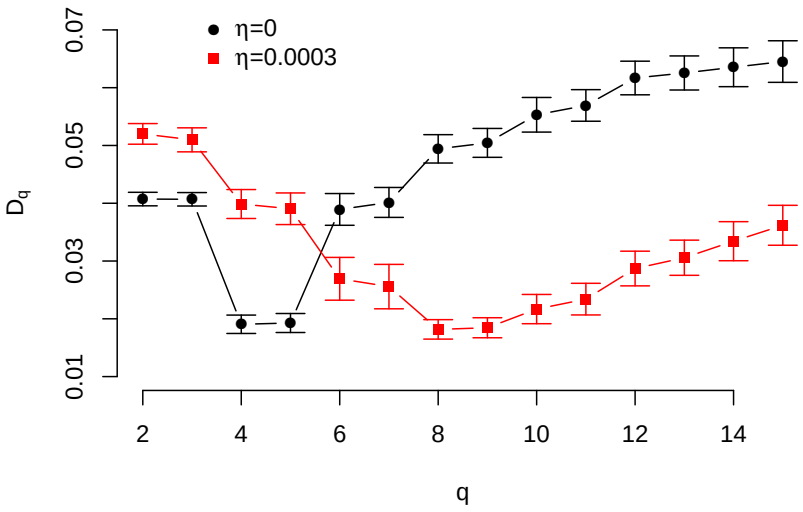

Figure 3.1: The degree of polynomials q versus the average Kolmogorov-Smirnov distance D_q with one standard deviation from 100 replications for $(p = 500, n = 30, \eta = 0)$ and $(p = 500, n = 30, \eta = 0.0003)$ using ridge inverse with $\lambda = 1e^{-08}$.

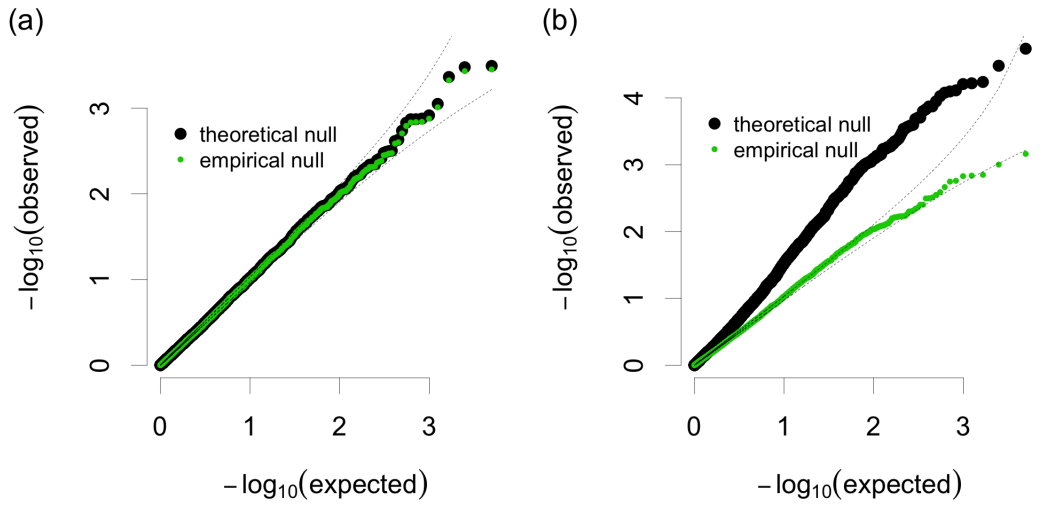

Figure 3.2: QQ-plots for p-values calculated using theoretical null distribution (black) or null distribution estimated by central matching method (green) against the expected uniform distribution on [0,1]. (a) $p = 100$ and $n = 1000$, (b) $p = 100$ and $n = 110$. The dotted lines are the 90% confidence limits of the expected values.

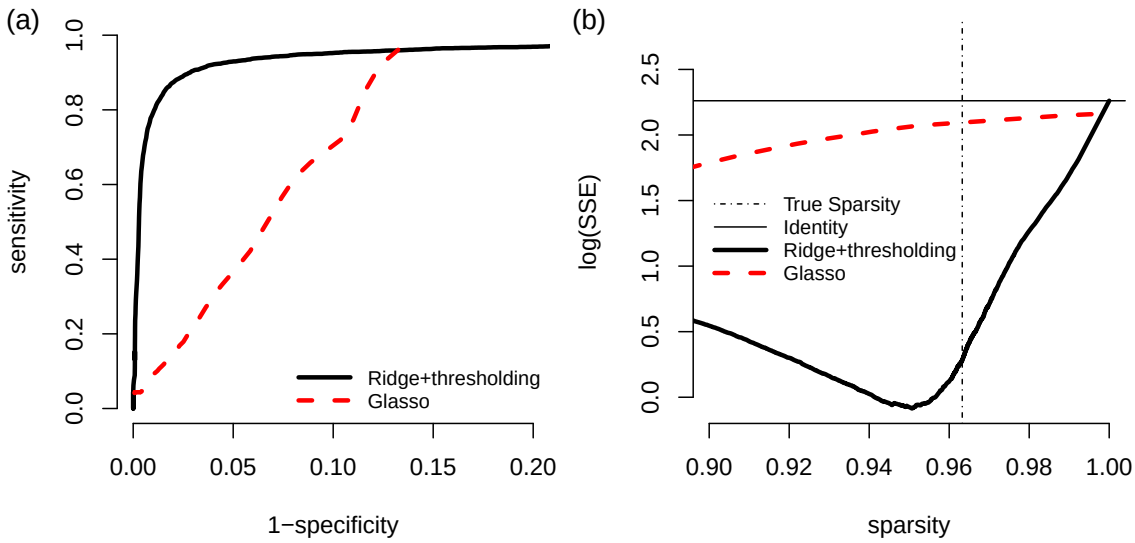

Figure 3.3: The ROC curve and SSE curve for $n = 100, p = 50$, and $|\mathbf{E}| = 45$. (a) ROC curve: 1-specificity versus sensitivity. (b) SSE curve: sparsity versus log(SSE). The horizontal black line is log(SSE) values when a $p \times p$ identity matrix is used and the vertical black line indicates the sparsity of the true network.

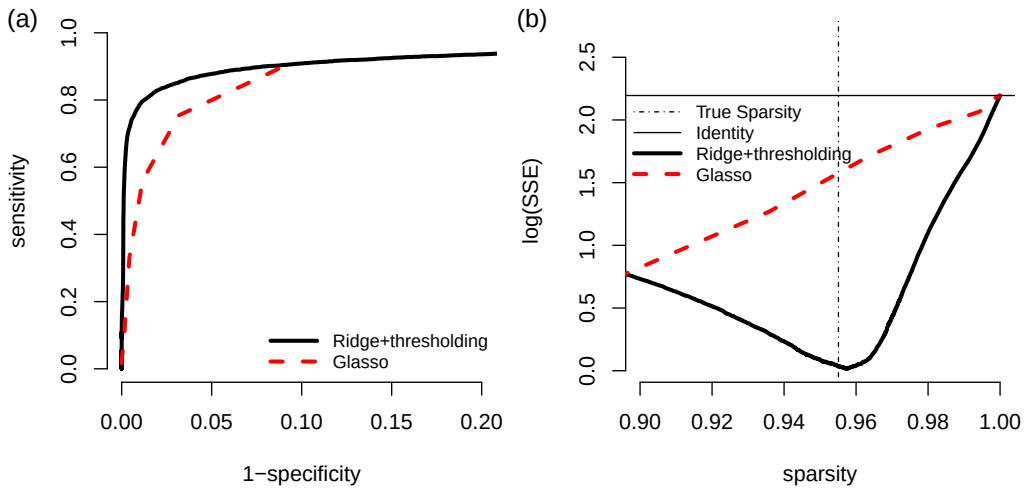

Figure 3.4: ROC curve and SSE curve for $n = 100, p = 50$, and $|\mathbf{E}| = 55$. (a) ROC curve: 1-specificity versus sensitivity. (b) SSE curve: sparsity versus log(SSE). The horizontal black line is log(SSE) values when a $p \times p$ identity matrix is used and the vertical black line indicates the sparsity of the true network.

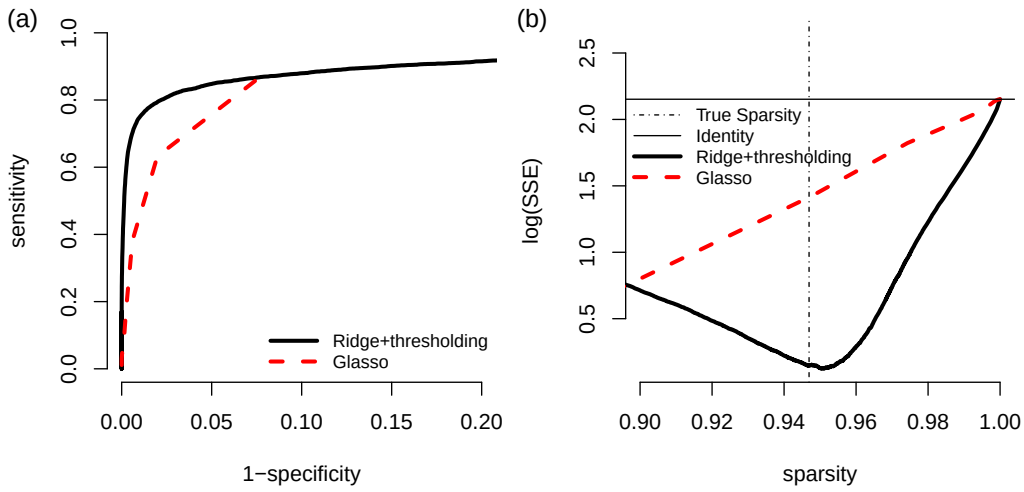

Figure 3.5: ROC curve and SSE curve for $n = 100, p = 50$, and $|\mathbf{E}| = 65$. (a) ROC curve: 1-specificity versus sensitivity. (b) SSE curve: sparsity versus log(SSE). The horizontal black line is log(SSE) values when a $p \times p$ identity matrix is used and the vertical black line indicates the sparsity of the true network.

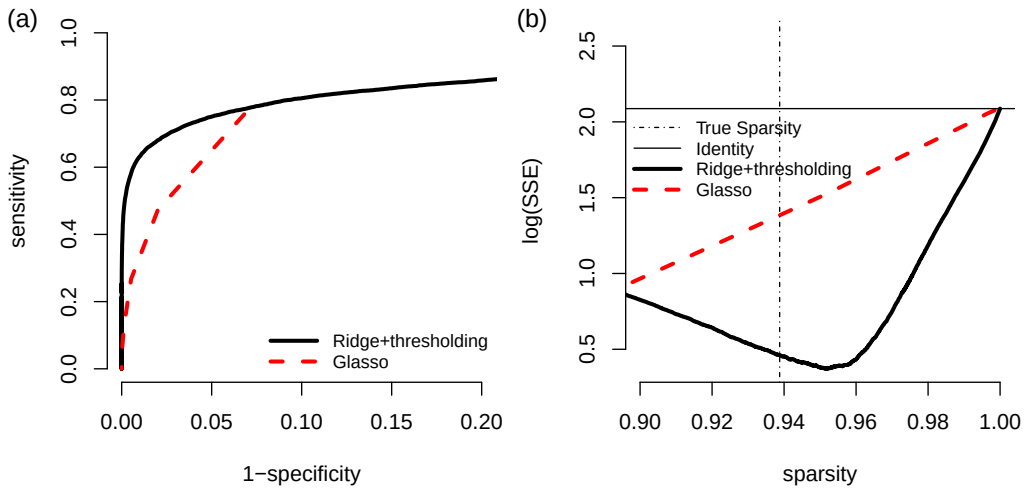

Figure 3.6: ROC curve and SSE curve for $n = 100, p = 50$, and $|\mathbf{E}| = 75$. (a) ROC curve: 1-specificity versus sensitivity. (b) SSE curve: sparsity versus log(SSE). The horizontal black line is log(SSE) values when a $p \times p$ identity matrix is used and the vertical black line indicates the sparsity of the true network.

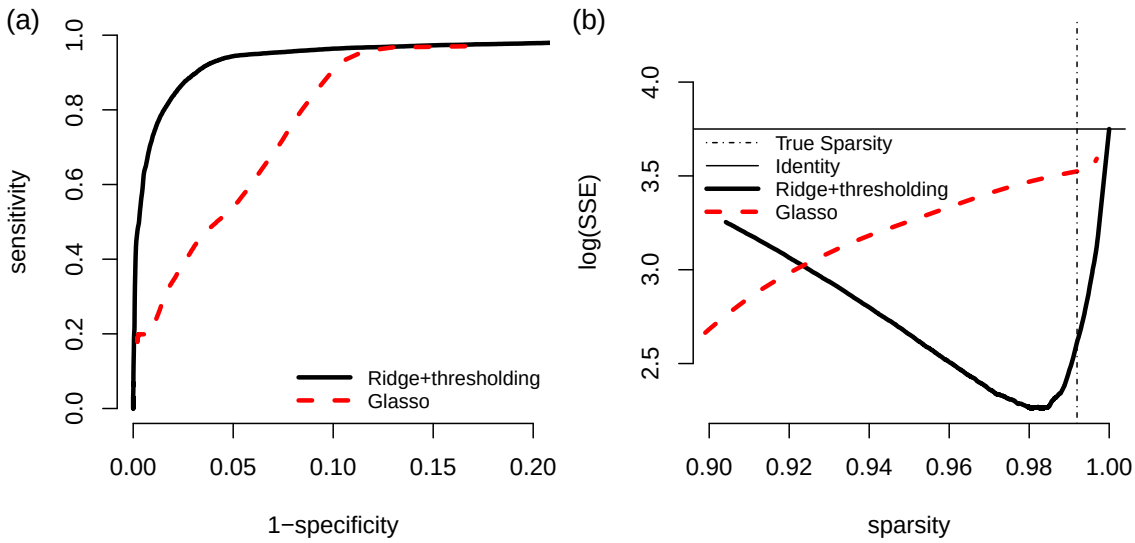

Figure 3.7: The ROC curve and SSE curve for $n = 100, p = 200$, and $|\mathbf{E}| = 160$. (a) ROC curve: 1-specificity versus sensitivity. (b) SSE curve: sparsity versus log(SSE). The horizontal black line is log(SSE) values when a $p \times p$ identity matrix is used and the vertical black line indicates the sparsity of the true network.

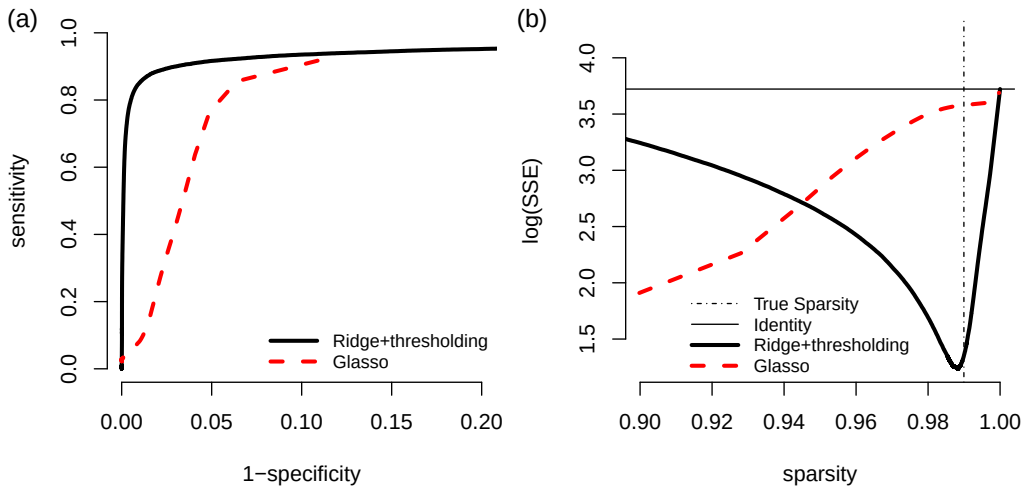

Figure 3.8: ROC curve and SSE curve for $n = 100, p = 200$, and $|\mathbf{E}| = 200$. (a) ROC curve: 1-specificity versus sensitivity. (b) SSE curve: sparsity versus log(SSE). The horizontal black line is log(SSE) values when a $p \times p$ identity matrix is used and the vertical black line indicates the sparsity of the true network.

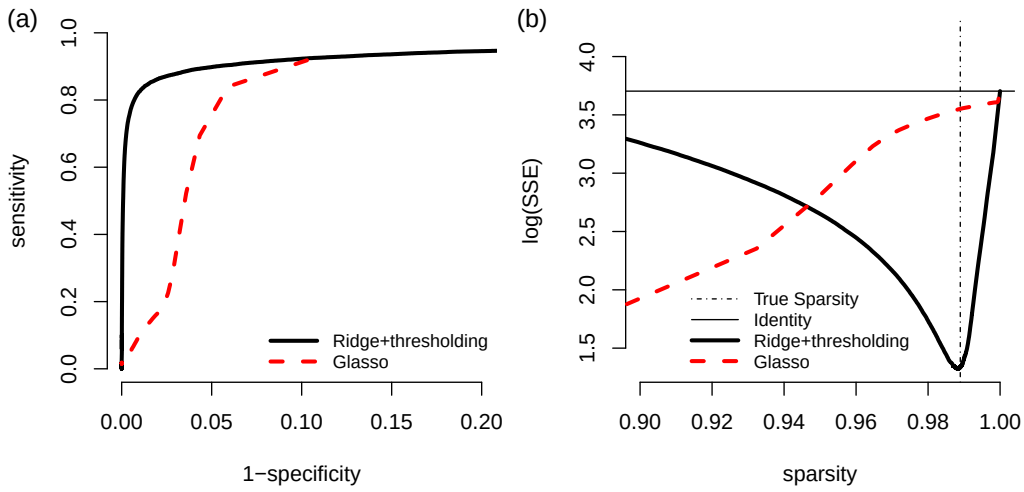

Figure 3.9: ROC curve and SSE curve for $n = 100, p = 200$, and $|\mathbf{E}| = 220$. (a) ROC curve: 1-specificity versus sensitivity. (b) SSE curve: sparsity versus log(SSE). The horizontal black line is log(SSE) values when a $p \times p$ identity matrix is used and the vertical black line indicates the sparsity of the true network.

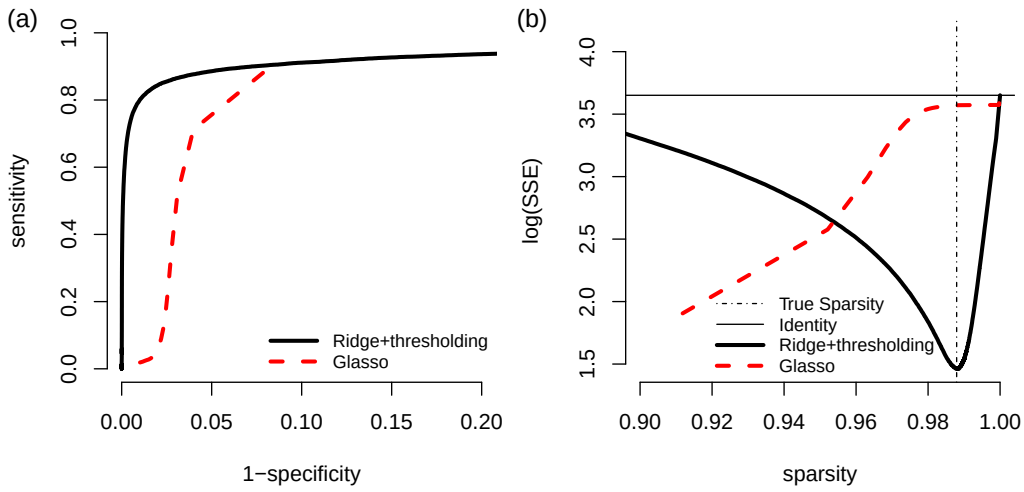

Figure 3.10: ROC curve and SSE curve for $n = 100, p = 200$, and $|\mathbf{E}| = 240$. (a) ROC curve: 1-specificity versus sensitivity. (b) SSE curve: sparsity versus log(SSE). The horizontal black line is log(SSE) values when a $p \times p$ identity matrix is used and the vertical black line indicates the sparsity of the true network.

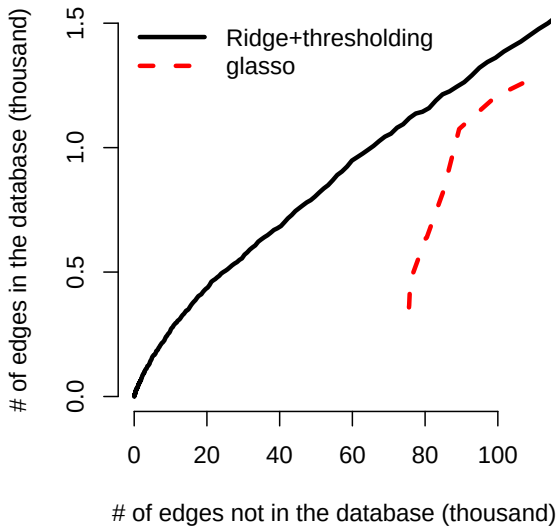

Figure 3.11: Comparing our method (Ridge+thresholding) with Glasso in terms partial correlation graph estimation by ROC curves, while the underlying true connections are defined as gene pairs belonging the the same cluster and their proteins having protein-protein interaction.

Chapter 4

PenPC: A Two-step Approach to Estimate the Skeletons of High Dimensional Directed Acyclic Graphs

4.1 Introduction

The relation of a set of random variables can be studied by graphical models, where vertices represent the variables and edges capture the relations among the variables [Lauritzen, 1996]. A particular class of graphs, the directed acyclic graphs (DAGs) (also known as Bayesian Network) have been well studied for its importance in causal inference [Pearl, 2009]. For example, in genomic studies, DAGs have been employed to study gene expression regulation [Friedman, 2004; Sachs et al., 2005; Zhang et al., 2010; Bonn et al., 2012]. In a DAG, all the edges are directed, and the direction of an edge implies a direct causal relation. There is no loop in a DAG, which is necessary to study causal relation [Spirtes et al., 2000]. When we remove the directions of all the edges in a DAG, the resulting undirected graph is the *skeleton* of the DAG.

Estimation of the skeleton of a DAG is of great importance. First, it is a crucial step towards estimation of the underlying DAG. Second, in many real data analyses where only observational data (instead of interventional data) are available, the DAG is not identifiable but the skeleton can be estimated; and previous studies have shown that causal effects can be assessed from the skeleton of a DAG [Maathuis et al., 2009,

2010]. Several methods have been developed to estimate DAGs or their skeletons [Heckerman et al., 1995; Spirtes et al., 2000; Chickering, 2003; Kalisch and Bühlmann, 2007], however most of them are only computationally feasible when the number of variables p is smaller or comparable to sample size n, with the exception of the PC-algorithm (named after its authors, Peter and Clark). Kalisch and Bühlmann [2007] proved that under some regularity conditions, the PC-algorithm consistently estimates the skeleton of sparse DAG for high-dimensional problems where $p = O(n^r)$ for $r > 0$. In this chapter, we proposed a new method named `PenPC` to address this challenging skeleton estimation problem. We proved the estimation consistency of `PenPC` under weaker regularity conditions for high dimensional settings of $p = O(n^r)$ or $p = O\left(\exp\{n^a\}\right)$. As verified by both simulation and real data analysis, `PenPC` provides more accurate estimates of the skeletons than the PC-algorithm.

The remaining parts of this chapter is organized as follows. In section 4.2, we give a brief review of Gaussian Graphical Models (GGMs), DAGs, and the conceptual advantage of our `PenPC` algorithm. Details of the `PenPC` algorithm is introduced in section 4.3 and its theoretical properties are presented in section 4.4. In section 4.5, we compare the performances of the `PenPC` and the PC algorithms by simulations. In section 4.6, we further evaluate the `PenPC` and the PC algorithms in real data analysis where causal effects estimated from observational data can be assessed by interventional data. In section 4.7, we observe order-dependency of `PenPC` algorithm and introduce order-independent `PenPC`. Finally, we conclude in section 4.8.

4.2 Review of Gaussian Graphical Models and DAGs

4.2.1 Gaussian Graphical Models (GGMs)

We consider a p-dimensional random vector $X = (X_1, ..., X_p)^{\mathrm{T}} \in \mathbb{R}^p$ following a multivariate normal distribution $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with unknown mean values $\boldsymbol{\mu}$ and a $p \times p$

non-singular covariance matrix $\mathbf{\Sigma}$. Let $\mathbf{\Omega} = [\omega_{ij}]_{p\times p} = \mathbf{\Sigma}^{-1}$ be the concentration matrix or precision matrix. Under multivariate normal assumption, $\omega_{ij} = 0$ if and only if X_i is independent with X_j given all other $p-2$ variables. Therefore $\mathbf{\Omega}$ is also known as partial covariance matrix. Let n be the sample size and denote the $n \times p$ observed data matrix by $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_p)$. Recently, a significant amount of works have been devoted to the estimation of $\mathbf{\Sigma}$ [Bickel and Levina, 2008; Levina et al., 2008; Rothman et al., 2008, 2009; Lam and Fan, 2009; Cai and Liu, 2011] or $\mathbf{\Omega}$ [Yuan and Lin, 2007; Rothman et al., 2008; Banerjee et al., 2008; Friedman et al., 2008; Fan et al., 2009; Yuan, 2010] from the observed data matrix $\mathbf{X}$ in high dimensional problems where p is much larger than n, see [Pourahmadi, 2011] for a recent review.

In this chapter, we are particularly interested in the identification of the non-zero entries of $\mathbf{\Omega}$, known as the covariance selection problem [Dempster, 1972]. It has been recognized that covariance selection and the estimation of concentration matrix are different problems [Meinshausen and Bühlmann, 2006; Yuan, 2010]. For example, the neighborhood selection method [Meinshausen and Bühlmann, 2006], which separately selects the neighbors of each vertex by a penalized regression with $p-1$ covariates, consistently estimates the nonzero elements of $\mathbf{\Omega}$, but only provides an approximate, instead of exact Penalized Maximum Likelihood Estimate (PMLE) of $\mathbf{\Omega}$ [Friedman et al., 2008].

Assuming that the variables of interest $X = (X_1, ..., X_p)^{\mathrm{T}}$ follow multivariate normal distribution, this covariance selection problem is equivalent to constructing a Gaussian Graphic Model (GGM). A GGM of X is an undirected graph $\mathcal{C} = (V, F)$ where V contains p vertices correspond to $X_1, ..., X_p$, and F contains all the undirected edges $i-j$ denoted by $(i, j) \in F$. There is an edge between vertices i and j if and only if X_i is dependent with X_j given all the other $p-2$ random variables, which is equivalent to $\omega_{ij} \neq 0$ in the concentration matrix $\mathbf{\Omega}$. A Gaussian Graphic Model is different from

the skeleton of a DAG because of v-structures. In a *v-structure* $X \to W \leftarrow Z$, X and Z are marginally independent or conditionally independent given the parents of X and Z, but given every set that contain W (a collision vertex) but not X or Z, X and Z are dependent with each other. For example, consider a sprinkler which is scheduled to spray at certain time every day. Either rain or the sprinkler may lead to the wet grass. Given the event that the grass is wet, there is a negative correlation between the event "sprinkler being on" and the event of rain [Pearl, 2009]. Other examples include the DAGs shown in Figure 4.1(a-d), where X and Z are not connected in skeleton, but they are connected in the corresponding Gaussian Graphic Models. Instances of the covariance and concentration matrices of the GGM in Figure 4.1(a) are shown in the Appendix II. The true network skeleton that there is no edge between X and Z can be identified by examining marginal correlations or conditional correlations. For example, $X \perp Z$ in Figure 4.1(a), $X \perp Z|Y$ in Figure 4.1(b), $X \perp Z|(Y, U)$ in Figure 4.1(c), and $X \perp Z|Y$ in Figure 4.1(d).

When the p variables are ordered by the underlying DAG's topology, such as $X_i \perp \{X_{i+1}, ..., X_p\} \mid \{X_1, ..., X_{i-1}\}$ for $i = 2, ..., p-1$, the problem of skeleton estimation is greatly simplified because a regression of X_i versus X_1, ..., X_{i-1} can be used to identify the true skeleton. Such a multiple regression won't be confused by v-structures because a common child of vertexes X_i and X_j will never appear as a covariate of the regression models using X_i or X_j as the response variable. In fact, Shojaie and Michailidis [2010] have shown that when the p variables are ordered by network topology, a neighborhood selection using the predecessors of each variable yields the exact PMLE of the concentration matrix rather than an approximation. However, in high-dimensional real data analysis problems, the topology order is rarely available. Throughout this chapter, we assume such an topology order is unknown.

4.2.2 Directed Acyclic Graph (DAGs)

A DAG of random variables $X_1, ..., X_p$ is a directed graph with no cycle (or loop). Specifically, a DAG can be denoted by $\mathcal{G} = (V, E)$, where V contains p vertices $1, 2, ..., p$ that correspond to $X_1, ..., X_p$, and E contains all the directed edges. In a DAG, a *path* of length n from i to j is a sequence $i = i_0 \to i_1 \to \cdots \to i_n = j$ of distinct vertices such that $(i_{l-1}, i_l) \in E$ for $l = 1, ..., n$. Given this path, i_{l-1} is a *parent* of i_l, i_l is a *child* of i_{l-1}, $i_0, i_1, ..., i_{l-1}$ are *ancestors* of i_l, and $i_{l+1}, ..., i_n$ are *descendants* of i_l. In a DAG, there is no path initiated from vertex i reaches i itself. This restriction of no cycle is necessary for causal inference. The adjacency set of vertices of j, denoted by $\texttt{adj}(j, \mathcal{G})$, are the vertices that are connected to j by an edge of any directionality.

A *chain* of length n from i to j is a sequence $i = i_0, i_1, \cdots, i_n = j$ of distinct vertices such that $i_{l-1} \to i_l$ or $i_l \to i_{l-1}$ for $l = 1, ..., n$. A DAG can graphically represent the conditional independence relationships among p variables by the following *d-separation* concept. A vertex set $\mathbf{S}$ block a chain $\texttt{p}$ if either (i) $\texttt{p}$ contains at least one arrow-emitting vertex that is in $\mathbf{S}$, or (ii) $\texttt{p}$ contains at least one collision vertex that is outside $\mathbf{S}$ and has no descendant of the collision vertex in $\mathbf{S}$. If $\mathbf{S}$ blocks all the chains from X to Y, it is said to "d-separate X and Y" [Pearl, 2009].

Not all the distributions can be faithfully represented by a DAG. A probability distribution $\mathcal{P}$ is *faithful* with respect to a DAG $\mathcal{G}$ if the conditional independence of $\mathcal{P}$ is equivalent to d-separation in $\mathcal{G}$. In this chapter, we assume that $X = (X_1, ..., X_p)^{\mathrm{T}} \in \mathbb{R}^p$ follow multivariate normal distribution. Among all the multivariate normal distributions associated with $\mathcal{G}$, the non-faithful ones form a Lebesgue null set [Meek, 1995b]. In the following discussions, we assume the faithfulness of the distributions.

A DAG is not identifiable from observational data, because conditional dependencies only determine the *skeleton* and *v-structures* of the graph [Pearl, 2009]. All the DAGs with the same skeleton and v-structures correspond to the same probability

distribution and they form an equivalence class, which can be described by a completed partially directed acyclic graph (CPDAG) [Chickering, 2002]. Identification of v-structures (hence a CPDAG), after skeleton estimation, only requires application of a set of deterministic rules, which is described in the Appendix II. Given a CPDAG, we can use the intervention calculus method developed by [Maathuis et al., 2009] to infer causal effects.

4.2.3 Constraint based approaches

In this section we will review constraint based methods to estimate the Markov equivalence class of a DAG. Under faithfulness assumption, the Inductive Causation (IC) algorithm aims at estimating the CPDAG of a DAG and the algorithm consists of three steps: (1) estimation of the skeleton by a set of conditional independence tests, (2) v-structure identification, and (3) completion of the PDAG obtained from (1) and (2) [Pearl, 2009]. The resulting graph is CPDAG which represents the Markov equivalence class of a DAG $\mathcal{G}$. After estimating skeletons using conditional independence tests, the steps (2) and (3) proceed by applying several deterministic rules described in [Pearl, 2009; Spirtes et al., 2000; Meek, 1995a; Chickering, 2002; Dor and Tarsi, 1992]. The estimation accuracy mostly depends on the first step, the sekeleton estimation.

Spirtes et al. [2000] describes various algorithms to estimate the skeleton. SGS algorithm starts from a complete undirected graph where any pair of vertices are connected then thins the graph by removing the edges $i - j$ such that Y_i and Y_j are conditionally independent given any subset in $V \setminus \{i, j\}$. PC algorithm thins the correlation graph by removing edges with first order conditional independence relations, thins again with second order conditional independence relations, and so on. The SGS algorithm relies on higher order conditional independence testings even for sparse graphs while for the PC algorithm, the set of variables conditioned on requires only be a subset of the set of

variables adjacent to one or the other variables tested. Subsequently, IG algorithm first estimate the undirected independence graph which is GGM under Gaussian assumption, then SGS algorithm is applied in each clique to exclude the false connections. As a variation of the IG algorithm, Spirtes et al. [2000] also suggested to apply PC algorithm in the second step.

In a high dimensional and sparse setting, Kalisch and Bühlmann [2007] proved uniform consistency of PC-algorithm when $p = O(n^a)$ for $a > 0$. Specifically, each test of conditional independence has certain probability of making a mistake, and they showed that under some regularity and sparsity conditions, the summation of these mistaken probabilities goes to 0. Using stability selection, Stekhoven et al. [2012] shows the improvement of IDA method in Maathuis et al. [2009] which provides estimated lower bounds of total causal effects based on the estimated CPDAG from PC-algotirhm. Colombo and Maathuis [2012] improved PC-algorithm by solving the order dependency from which the resulting skeleton depends on the variable ordering of the input data. It is called *PC-stable* algorithm.

Our `PenPC` algorithm has two steps. It first adapts neighborhood selection method to estimate zero structure of the concentration matrix, which gives a GGM under multivariate normal distribution assumption, and then apply a modified PC-stable algorithm on the estimated GGM to remove false positive edges that connect the parents of a common child due to v-structures. Our two step approach has the same spirit as the IG algorithm and Schmidt et al. [2007] used the neighborhood selection approach to estimate topological ordering. We employ the log penalty ($p_{\lambda,\tau}(|b|) = \lambda \log(|b|+\tau)$), one of the folded concave penalties [Fan and Lv, 2011], for neighborhood selection, which significantly improves the accuracy of resulting GGM. In the second step of removing false positives, because we only aim to identify a particular class of false positives due to v-structures, the number of tests, hence the cumulative mistaken probabilities, are

significantly reduced. After adding up the uncertainty of neighborhood selection in the first step and the cumulative mistaken probabilities in the second step, we can still obtain consistent estimate of the skeleton while allowing $p = O(n^r)$ or $p = O(\exp\{n^a\})$. Previous works on network skeleton estimation have assumed traditional random graph model where all the vertexes have the same expected number of connections [Kalisch and Bühlmann, 2007]. However, many real networks are scale-free graphs where a few vertexes may have much larger number of connections than the other vertices. We show both theoretically and empirically that `PenPC` algorithm performs well for such scale-free DAGs.

4.3 Methods

Let $V = \{1, \ldots, p\}$ be the vertex indices. Our `PenPC` algorithm proceeds in two steps: (1) estimation of a GGM $\mathcal{C}_{\mathcal{G}} = (V, F_{\mathcal{G}})$ by neighborhood selection, and (2) application of a modified PC algorithm to remove false connections.

Step 1. (Neighborhood Selection) We first select the neighborhood of vertex i by a penalized regression with X_i as response variable and all the other variables corresponding to vertices $V \setminus \{i\}$ as covariates:

$$\hat{\boldsymbol{b}}_i = \underset{\boldsymbol{b}_i \in \mathbb{R}^{p-1}}{\arg\min} \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{X}_{-i}\boldsymbol{b}_i)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{X}_{-i}\boldsymbol{b}_i) + n\sum_{j \neq i} p_{\boldsymbol{\theta}}(|b_{i,j}|) \tag{4.1}$$

where $\boldsymbol{x}_i$ is $n \times 1$ vector for n measurements of variable X_i, $\boldsymbol{X}_{-i}$ is $n \times (p-1)$ matrix for n measurements of the remaining $p-1$ covariates, $\boldsymbol{b}_i = (b_{i,1}, ..., b_{i,i-1}, b_{i,i+1}, ..., b_{i,p})^{\mathrm{T}}$ and $p_{\boldsymbol{\theta}}(|b_{i,j}|)$ denotes a penalty function with tuning parameters $\boldsymbol{\theta}$. We consider a class of folded concave penalty functions satisfying the following condition:

> **Condition 1:** Penalty function $p_{\boldsymbol{\theta}}(t)$ is increasing and concave in $t \in [0, \infty)$ given $\boldsymbol{\theta}$ and has continuous derivative $p'_{\boldsymbol{\theta}}(t)$ with $p'_{\boldsymbol{\theta}}(0+) > 0$.

This is a generalization of the Condition 1 in [Fan and Lv, 2011] for any dimensionality of $\boldsymbol{\theta}$. The penalty $p_{\boldsymbol{\theta}}$ at $\mathbf{v} = (v_1, ..., v_r)^{\mathrm{T}} \in \mathbb{R}^r$ with $\|\mathbf{v}\|_0 = r$ has nonnegative local concavity $\kappa(p_{\boldsymbol{\theta}}; \mathbf{v}) \geq 0$ where

$$\kappa(p_{\boldsymbol{\theta}}; \mathbf{v}) = \lim_{\epsilon \to 0+} \max_{1 \leq j \leq r} \sup_{t_1 < t_2 \in (|v_j| - \epsilon, |v_j| + \epsilon)} -\frac{p'_{\boldsymbol{\theta}}(t_2) - p'_{\boldsymbol{\theta}}(t_1)}{t_2 - t_1}, \tag{4.2}$$

and $\|\cdot\|_0$ is the L_0 norm of a vector [Fan and Lv, 2011]. Specifically, we employed the log penalty ($p_{\lambda,\tau}(|b|) = \lambda \log(|b| + \tau)$) in this chapter. After p penalized regressions for each of the p variables, we construct the GGM as follows. Start with a graph with only p vertices but no edge, and add an edge between vertices i and j if $\hat{b}_{ij} \neq 0$ or $\hat{b}_{ji} \neq 0$, where $i, j \in V$ and $i \neq j$.

Step 2. (Modified PC-algorithm) We apply a modified PC algorithm to remove the false edges due to co-parent relationships. Denote $\mathtt{adj}(i, \mathcal{C}_{\mathcal{G}})$ as the adjacent vertices of i in the GGM $\mathcal{C}_{\mathcal{G}} = (V, F_{\mathcal{G}})$. Let $\mathtt{adj}(i, j, \mathcal{C}_{\mathcal{G}}) = \mathtt{adj}(i, \mathcal{C}_{\mathcal{G}}) \bigcap \mathtt{adj}(j, \mathcal{C}_{\mathcal{G}})$. The subgraph of $\mathcal{C}_{\mathcal{G}}$ on the set of vertices $S \subseteq V$ is denoted by $\mathcal{C}_{\mathcal{G}}(S)$. Let $\mathtt{Con}(v, \mathcal{C}_{\mathcal{G}}(S))$ for $v \in S$ be the set of vertices connected to v by any length of chains in $\mathcal{C}_{\mathcal{G}}(S)$, including v itself. For a pair of vertices i and j connected in $\mathcal{C}_{\mathcal{G}}$, we test whether they are independent conditioning on each set in

$$\boldsymbol{\Pi}_{i,j} \equiv \left\{ \left[\mathtt{adj}(i, \mathcal{C}_{\mathcal{G}}) \bigcup \mathtt{adj}(j, \mathcal{C}_{\mathcal{G}}) \right] \setminus \Gamma : \ \Gamma \subseteq \boldsymbol{\Gamma}_{i,j} \right\} \setminus \{i, j\}, \tag{4.3}$$

where $\setminus$ indicates set difference and

$$\boldsymbol{\Gamma}_{i,j} = \left[\bigcup_{v \in \mathtt{adj}(i,j,\mathcal{C}_{\mathcal{G}})} \mathtt{Con}\big(v, \mathcal{C}_{\mathcal{G}}(V \setminus \{i, j\})\big) \right] \bigcap \left[\mathtt{adj}(i, \mathcal{C}_{\mathcal{G}}) \bigcup \mathtt{adj}(j, \mathcal{C}_{\mathcal{G}}) \right].$$

Note that $\boldsymbol{\Pi}_{i,j}$ is a collection of sets. Each set belonging to $\boldsymbol{\Pi}_{i,j}$ corresponds to a subset of $\boldsymbol{\Gamma}_{i,j}$. We test the conditional independence of X_i and X_j given $\mathcal{K} \in \boldsymbol{\Pi}_{i,j}$ using

Fisher transformation of partial correlation. Specifically, denote the partial correlation between X_i and X_j given $\mathcal{K} \in \mathbf{\Pi}_{i,j}$ by $\rho_{i,j|\mathcal{K}}$. With the significance level α, we reject the null hypothesis $H_0 : \rho_{i,j|\mathcal{K}} = 0$ against the alternative hypothesis $H_a : \rho_{i,j|\mathcal{K}} \neq 0$ for $\mathcal{K} \in \mathbf{\Pi}_{i,j}$ if $\sqrt{n - |\mathcal{K}| - 3}\hat{z}_{i,j|\mathcal{K}} > \Phi^{-1}(1-\alpha/2)$, where $\hat{z}_{i,j|\mathcal{K}} = 0.5\log((1+\hat{\rho}_{i,j|\mathcal{K}})/(1-\hat{\rho}_{i,j|\mathcal{K}}))$ and $\Phi(\cdot)$ is the cdf of $N(0,1)$.

The details of the step 2 of `PenPC` algorithm is described in the Supplementary Materials, Section 5.6. Here we give a brief description of its rationale. If two vertices i and j are not connected in the skeleton, but connected in the GGM, it must be due to a v-structure (see Lemma 2 of Section 4). Then i and j are independent or conditional independent if the conditional set includes all the adjacent vertices of i or j but excludes the common children of i and j plus all descendants of the common children. Therefore all the common children of i and j and those descendants must be included in $\mathbf{\Gamma}_{i,j}$. For example to test X and Y in Figure 4.1 (d) $\mathbf{\Gamma}_{i,j}$ must include the common children W and its descendants U and V. See Lemma 3 of Section 4.4.1 for a rigorous description.

The final output of `PenPC` algorithm is the estimated skeleton and separation sets $S(i,j)$ for all (i,j). The separate sets are needed for causal effect estimation. If vertices i and j are not connected in the GGM (then they won't be connected in the skeleton), their separation set is $V \setminus \{i,j\}$. If i and j are connected in both the GGM and the skeleton, there is no separation set. If i and j are connected in the GGM, but not the skeleton, the separation set $S(i,j)$ is a set belongs to $\mathbf{\Pi}_{i,j}$, such that the test $i \perp j \mid S(i,j)$ gives affirmative conclusion. Given the skeleton and the separation sets, causal effects can be assessed using function `idaFast` of R package pcalg [Kalisch et al., 2012].

4.4 Theoretical Properties

4.4.1 Fixed Graphs

We denote the L_2 and L_∞ norm of a matrix or a vector by $\|\cdot\|_2$ and $\|\cdot\|_\infty$. The L_2 norm of a square matrix is the maximum eigenvalue of the matrix. The L_∞ norm of a matrix is the maximum of the L_1 norm of each row. The L_∞ norm of a vector is the maximum of the absolute values of its elements. In this section we study high dimensional behavior where p grows as a function of sample size n. Thus denote $\mathcal{G}_n = (V_n, E_n)$ and $\mathcal{C}_{\mathcal{G}_n} = (V_n, F_n)$ with $|V_n| = p_n$ as a DAG and the GGM specified by the moral graph of $\mathcal{G}_n$, respectively. We also denote the skeleton of $\mathcal{G}_n$ by $\mathcal{G}_n^u = (V_n, E_n^u)$ where $(a, b) \in E_n^u \Leftrightarrow (a, b) \in E_n$ or $(b, a) \in E_n$. The following conditions are needed for the consistency of the `PenPC` algorithm.

(A1) The distribution of the $X \in \mathbb{R}^{p_n}$ is multivariate Gaussian and is faithful to the DAG $\mathcal{G}_n$ for all n, where $p_n \leq O\left(\exp\{n^a\}\right)$ with $a \in [0, 1)$. Recall that the $n \times p_n$ observed data matrix is denoted by $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_{p_n})$. Without loss of generality, we assume each column $\boldsymbol{x}_j$ $(1 \leq j \leq p_n)$ has been standardized to have mean 0 and $\boldsymbol{x}_j^{\mathrm{T}} \boldsymbol{x}_j = n$.

(A2) Let q_n be the maximum degree of $\mathcal{C}_{\mathcal{G}_n}$, i.e., $q_n = \max_{1 \leq j \leq p_n} |\mathtt{adj}(j, \mathcal{C}_{\mathcal{G}_n})|$. Suppose $q_n \leq O(n^b)$ for some $0 \leq b < 1$. By the following Lemma 2, $\max_{1 \leq j \leq p_n} |\mathtt{adj}(j, \mathcal{G}_n)| \leq \max_{1 \leq j \leq p_n} |\mathtt{adj}(j, \mathcal{C}_{\mathcal{G}_n})| = q_n$.

(A3) Denote the partial correlations between X_i and X_j given a set of variables $\{X_r : r \in \mathcal{K}\}$ for some set $\mathcal{K} \subseteq V_n \setminus \{i, j\}$ by $\rho_{i,j|\mathcal{K}}$. The absolute values of $\rho_{i,j|\mathcal{K}}$'s are bounded from below and above:

$$\inf_{i,j,\mathcal{K}} \left\{ \left| \rho_{i,j|\mathcal{K}} \right| : \rho_{i,j|\mathcal{K}} \neq 0 \right\} \geq c_n, \text{ and } \sup_{n,i,j,\mathcal{K}} \left| \rho_{i,j|\mathcal{K}} \right| \leq M < 1,$$

where $c_n = O(n^{-d_1})$ for some $0 < d_1 < 1/2$.

(A4) For any vertex i, denote the observed data of the variables within and outside of $\texttt{adj}(i, \mathcal{C}_{\mathcal{G}_n})$ (but not including $\boldsymbol{x}_i$) by $\boldsymbol{\mathcal{X}}_{i1}$ and $\boldsymbol{\mathcal{X}}_{i2}$, respectively. Dependence among $\texttt{adj}(i, \mathcal{C}_{\mathcal{G}_n})$ is restricted such that for any i, $\|(\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1})^{-1}\|_\infty = O(n^{-1+s_0})$, where $0 \leq s_0 < (1-a)/2$.

(A5) Let $\delta_n = (1/2)\inf_{i,j}\{|b_{i,j}| : b_{i,j} \neq 0\}$, where $\delta_n \geq O(n^{-d_2})$ for some $0 < d_2 < (1-a)/2 - s_0$. The dependence between $\boldsymbol{\mathcal{X}}_{i1}$ and $\boldsymbol{\mathcal{X}}_{i2}$ is restricted by

$$\left\|\boldsymbol{\mathcal{X}}_{i2}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1}(\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1})^{-1}\right\|_\infty \leq \min(Kp'_{\boldsymbol{\theta}}(0+)/p'_{\boldsymbol{\theta}}(\delta_n), O(n^b)),$$

for $0 < K < 1$ and b in (A2).

(A6) $p'_{\boldsymbol{\theta}}(\delta_n) \ll n^{-d_2-s_0}$, $p'_{\boldsymbol{\theta}}(0+) \gg n^{-1/2+a/2+b}\sqrt{\log n}$, and $\max_{i\in V_n}\|(\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1})^{-1}\|_2 \leq 1/(n\kappa_0)$ where $\kappa_0 = \max_i \max_{\boldsymbol{\beta}\in\mathcal{N}_i}\kappa(p_{\boldsymbol{\theta}}, \boldsymbol{\beta}_1)$ with $\kappa(p_{\boldsymbol{\theta}}, \cdot)$ defined in (4.2) and $\mathcal{N}_i$ defined in equation (S5.18) of the Supplementary Materials.

We adapt (A1)-(A3) from [Kalisch and Bühlmann, 2007] to prove uniform consistency of step 2 of `PenPC` algorithm from known $\mathcal{C}_{\mathcal{G}_n}$. However (A1) allows higher dimensionality p_n in the exponential order of n. The sparseness assumption (A2) will be replaced by tighter assumptions for two specific random graph models later. Assumption (A4)-(A6) are needed to achieve the uniform oracle property of the non-concave penalized regressions. Thus, they ensure that the step 1 of `PenPC` can recover the graphical structure of partial correlation matrix. The following Lemma 1 claims that the support of the regression coefficients is the same as that of the concentration matrix. Therefore, we can use the regression model to detect the support the partial correlation matrix $\mathcal{C}_{\mathcal{G}_n}$. Denote X_{-i} as a $p_n - 1$ dimension random vector when X_i is excluded from X. $\boldsymbol{\Sigma}_{ab}$ and $\boldsymbol{\Omega}_{ab}$ are the sub-matrices of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Omega}$ corresponding to

random vectors X_a and X_b for $a, b \subseteq V_n$.

Lemma 1. *Suppose* $X = (X_1, ..., X_p)^{\mathrm{T}} \sim \mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *and* $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$. *Then*

$$X_i = X_{-i}^{\mathrm{T}} \boldsymbol{b}_i + \epsilon_i,$$

where $\boldsymbol{b}_i = -\sigma_i^2 \boldsymbol{\Omega}_{-i,i}$, *and* $\epsilon_i \sim \mathrm{N}(0, \sigma_i^2)$, *with* $\sigma_i^2 = \boldsymbol{\Sigma}_{ii} - \boldsymbol{\Sigma}_{i,-i}(\boldsymbol{\Sigma}_{-i,-i})^{-1}\boldsymbol{\Sigma}_{-i,i}$.

The proof of Lemma 1 is in the Appendix II which is similar to [Lauritzen, 1996]. Consider the neighborhood selection problem for one of the variables versus all the other variables. Let $\mathcal{S}_i = \mathrm{supp}(\boldsymbol{b}_i)$ be the support of the true regression coefficient $\boldsymbol{b}_i$ with the size $|\mathcal{S}_i| = s_i$. From Lemma 1, the degree of vertex i in $\mathcal{C}_{\mathcal{G}_n}$ is s_i. Recall that in assumption (A4) $\boldsymbol{\mathcal{X}}_{i1}$ and $\boldsymbol{\mathcal{X}}_{i2}$ are denoted by the observed data of the variables corresponding to $\mathcal{S}_i \subseteq V_n \setminus \{i\}$ and its complement, $\mathcal{S}_i^c = V_n \setminus (\mathcal{S}_i \bigcup \{i\})$. Similarly $\boldsymbol{b}_{i1}$ and $\hat{\boldsymbol{b}}_{i1}$ are respectively the sub-vectors of $\boldsymbol{b}_i$ and $\hat{\boldsymbol{b}}_i$ formed by $\mathcal{S}_i$.

Theorem 2. *Given Assumptions (A1), (A4)-(A6), with probability at least* $1 - C\exp\{n^a - n^a \log(n)\}$ *for a constant* $0 < C < \infty$, *there exists a local minimizer* $\hat{\boldsymbol{b}}_i = (\hat{\boldsymbol{b}}_{i1}, \hat{\boldsymbol{b}}_{i2})^{\mathrm{T}}$ *that satisfies the following conditions: for any* $i = 1, \ldots, p_n$,

(a) Sparsity: $\mathbb{P}(\hat{\boldsymbol{b}}_{i2} = \mathbf{0}) \to 1$.

(b) L_∞ *loss:* $\|\hat{\boldsymbol{b}}_{i1} - \boldsymbol{b}_{i1}\|_\infty = o(n^{-d_2})$, *where* d_2 *is defined in (A5).*

The proof is in the Appendix II. Under assumption (A1), the dimensionality p_n is allowed to grow up to exponentially fast with sample size n. The value of d_2 can be as large as $1/2$ depending on the lower bound of nonzero partial correlation signals (given all the other $p_n - 2$ variables).

Corollary 1 is a simple extension from Theorem 2. It characterizes the uniform oracle property for p_n penalized regression models which estimate the GGM $\mathcal{C}_{\mathcal{G}_n}$. Denote $\hat{\mathcal{C}}_{\mathcal{G}_n}(\boldsymbol{\theta})$ as the estimate of $\mathcal{C}_{\mathcal{G}_n}$ by the neighborhood selection, where $\boldsymbol{\theta}$ are tuning

parameters of the penalty function.

Corollary 1. *Given Assumption (A1), (A4)-(A6),*

$$\mathbb{P}(\hat{\mathcal{C}}_{\mathcal{G}_n}(\boldsymbol{\theta}) = \mathcal{C}_{\mathcal{G}_n}) \geq 1 - C\exp\{2n^a - n^a\log(n)\}$$

for a constant $0 < C < \infty$.

Lemma 2. *Assume (A1). The set of edges* F_n *of* $\mathcal{C}_{\mathcal{G}_n}$ *includes all edges* E_n^u *of* $\mathcal{G}_n^u$ *plus co-parent relationship in* $\mathcal{G}_n$.

This lemma 2 is proved in Lemma 3.21 of Lauritzen [1996] when recursive factorization of the joint distribution of X is assumed according to the directed graph $\mathcal{G}_n$.

Lemma 3. *Assume (A1). If* $(i,j) \in F_n$ *of* $\mathcal{C}_{\mathcal{G}_n}$ *but* $(i,j) \notin E_n^u$ *of* $\mathcal{G}_n^u$*, the conditioning set* $\mathbf{\Pi}_{i,j}$ *in (4.3) includes at least one set which d-separates vertices* i *and* j *in* $\mathcal{G}$.

Lemma 2 and Lemma 3 provide the theoretical justifications for using GGM as a starting point of our modified PC-algorithm. The proofs are in the Appendix II. Lemma 2 shows that if we have a perfect estimation of the partial covariance matrix, we can recover all the edges in the skeleton with no false negatives, but some false positives: the co-parent relationships. Lemma 3 presents that we can remove false positive connection between i and j due to co-parent relationship by examining partial correlation conditioning on some set in $\mathbf{\Pi}_{i,j}$.

Next we discuss the theoretical property of a modified PC algorithm (step 2 of the `PenPC` algorithm presented in section 3) given a perfect estimation of GGM. Later we will show that the summation of mistaken probabilities of GGM estimation and skeleton estimation given GGM goes to 0 as $n \to \infty$.

Theorem 3. *Let* $\hat{\mathcal{G}}_n^u(\alpha_n)$ *be the estimates of* $\mathcal{G}_n^u$ *from the second step of the* `PenPC` *algorithm, given a perfect estimation of GGM. Assume (A1)-(A3) with* $0 < d_1 <$

$\min((1-a)/2, (1-b)/2)$. *There exists an* $\alpha_n \to 0$, *such that*

$$\mathbb{P}\left[\hat{\mathcal{G}}_n^u(\alpha_n) = \mathcal{G}_n^u\right] = 1 - O\left(\exp\{-Cn^{1-2d_1}\}\right) \to 1,$$

where $0 < C < \infty$ *is a constant, and* α_n *is the p-value threshold for testing whether a partial correlation is 0.*

The proof is in the Appendix II. Similar theorem has been proved in [Kalisch and Bühlmann, 2007] with p_n at polynomial order of n. By exploiting accuracy estimation of GGM, we extend the theorem to $p_n = O(\exp\{n^a\})$ case. Corollary 2 provides the combined error of step 1 and step 2 of `PenPC` algorithm as a simple extension of Corollary 1 and Theorem 3.

Corollary 2. *Let* $\hat{\mathcal{G}}_n^u(\boldsymbol{\theta}, \alpha_n)$ *be the estimates of* $\mathcal{G}_n^u$ *from the two step approach* `PenPC` *algorithm. Assume (A1)-(A6) with* $0 < d_1 < \min((1-a)/2, (1-b)/2)$. *There exists an* $\alpha_n \to 0$, *such that*

$$\mathbb{P}\left[\hat{\mathcal{G}}_n^u(\boldsymbol{\theta}, \alpha_n) = \mathcal{G}_n^u\right] = 1 - O\left(\exp\{-Cn^{1-2d_1}\}\right) \to 1,$$

where $0 < C < \infty$ *is a constant.*

4.4.2 Random Graphs

Under certain conditions, the theoretical results could also be extended to two commonly used models for random graphs: Erdős and Rényi (ER) Model [Erdős and Rényi, 1960] and Barabási and Albert (BA) Model [Barabási and Albert, 1999]. In general, assumption (A2) no longer holds for random graphs. However, based on the proof in the Appendix II, it is easy to see that assumption (A2) can be relaxed to (A2').

(A2') Let $q_n = \max_{1 \le j \le p_n} |\texttt{adj}(j, \mathcal{C}_{\mathcal{G}_n})|$. Assume

$$\mathbb{P}\{q_n \le O(n^b)\} = 1, \quad \text{for some } 0 \le b < 1.$$

It is then suffices to show assumption (A2') holds. We also discuss the value of b in the assumption, which will affect the minimum effect size of partial correlations in assumption (A3) and the convergence probability in Theorem 2 and Corollary 1.

Erdős and Rényi (ER) Model

The ER model constructs a graph $G(p_n, p_E)$ of p_n vertices by connecting vertices randomly. Each edge is included in the graph with probability p_E independent from all other edges. By law of large numbers, such vertex is almost surely connected to $(p_n - 1)p_E$ edges. Let M_n be the maximal degree of the graph. Erdős and Rényi [1960] proved the following results about M_n.

Lemma 4. *In the graph $G(p_n, p_E)$ following the ER model, the maximal degree M_n almost surely converges to m_n, where*

$$m_n = \begin{cases} O(\log p_n), & \text{if } p_n p_E < 1, \\ p_n^{2/3}, & \text{if } p_n p_E = 1, \\ O(p_n), & \text{if } \lim_{p_n \to \infty} p_n p_E = c > 1. \end{cases}$$

When $p_n = O\{\exp(n^a)\}$, by Lemma 4, assumption (A2') holds immediately if $p_n p_E < 1$ and $b \ge a$. When $p_n p_E \ge 1$, our proof cannot handle the general case $p_n = O\{\exp(n^a)\}$. However, when the number of vertices is of the polynomial order of n, assumption (A2') may still hold. In particular, suppose $p_n = O(n^r)$. When $p_n p_E < 1$, assumption (A2') holds for any $b \in [0, 1)$. When $p_n p_E = 1$, assumption (A2') holds if $b \ge 2r/3$. When

$p_n p_E \to c > 1$, assumption (A2') holds if $r < 1$ and $b \geq r$.

Barabási and Albert (BA) Model

The BA model is used to generate scale free graphs whose degree distribution follows a power law: $\mathbb{P}(\nu) = \gamma_0 \nu^{-\gamma_1}$, with a normalizing constant γ_0 and a exponent γ_1. Specifically, BA model generates a graph by adding vertices into the graph over time and when each new vertex is introduced into the graph, it is connected with larger probability to the existing vertices with larger number of connections. Since the distribution does not depend on the size of the network (or time), the graph organizes itself into a scale free state [Barabási and Albert, 1999]. Móri [2005] showed that M_n almost surely converges to $O(p^{1/2})$. Thus, assumption (A2') holds for the case $p_n = O(n^r)$ with $b \leq r/2$.

4.5 Simulation Studies

We evaluate the performance of the `PenPC`-algorithm and the PC-algorithm in terms of sensitivity and specificity of skeleton estimation using DAGs simulated by the ER model or the BA model. Following Kalisch and Bühlmann [2007], we simulate DAGs of p vertices by the ER model as follows. First we assume the p vertices are ordered so that if $i < j$, vertex i can only be the parent rather than child of vertex j. Then for any vertex pair (i, j) where $i < j$, we add an edge $i \to j$ with probability p_E. For the BA model, the DAGs are simulated following Barabási and Albert [1999]. We start with a vertex with no edge in the beginning. Then a new vertex is added in each step and directed edges are added so that they start from the new vertex and point to some of the existing vertices. Specifically, in the $(t+1)$-th step, the new vertex is connected to a existing vertex repeatedly e times and the probability to connect to vertex $1 \leq i \leq t$ is $\nu_i^t / \sum_j \nu_j^t$ where $\nu_i^t = |\mathtt{adj}(i, \mathcal{G}^t)|$, and $\mathcal{G}^t$ is the DAG at the t-th step, right before adding

the new vertex. After p time steps, we have a DAG denoted by $\mathcal{G} = (V, E)$ with $|V| = p$ and $|E| \leq (p-1)e$. The inequality for $|E|$ is from the possibility that the new vertex is connected to the same old vertex when $e > 1$. Figure 4.2 displays the distribution of the degrees ν from simulated DAGs under ER model ($p = 1000$ and $p_E = 2/p$) and BA model ($p = 1000$ and $e = 1$). The probability of finding a highly connected vertex decreases exponentially with ν for the graphs generated by the ER model (left panel). However, for the graph generated by the BA model, highly connected vertices with large ν have relatively large chance of occurring (right panel). The subplot of right panel displays a linear relation between degree and degree probability in log-log scale, which confirms the scale-free property of the graph generated by the BA model. The BA model with $e = 2$ is displayed in Figure 4.3.

After constructing the DAGs, the observed data were simulated by structure equation under normal assumption. For example, let $\boldsymbol{x}_j$ be the n observed values for variable X_j, and denote the parents of X_j by $\mathtt{pa}_j$, then $\boldsymbol{x}_j = \sum_{k \in \mathtt{pa}_j} b_{jk} \boldsymbol{x}_k + \epsilon_j$, where $\epsilon_j \sim \mathcal{N}(0, \sigma^2 I_{n \times n})$. In our simulations, all b_{jk}'s and σ^2 are set to be 1. Our simulation settings are displayed in Table 4.1. For either ER or BA model, we consider low dimension setting where $p = 11, n = 100$ and high-dimension settings where $p = 100, n = 30$ and $p = 1000, n = 300$ with various sparsity levels determined by P_E for ER model and e for BA model. The results for all the simulation results are displayed in Figure 4.4-Figure 4.17. There are three tuning parameters in `PenPC`: λ and τ for the penalty function and α, which is the p-value cutoff used by the PC algorithm or our modified PC algorithm to declare conditional independence. We choose λ and τ by extended BIC [Chen and Chen, 2008], and examine the results of PC or `PenPC` across various values of α. For example, in the upper panels of Figure 4.10, we show the performances of three methods: PC (PC-algorithm), Pen (penalized regression only), and `PenPC` when $\alpha = 0.01$ and the skeleton is simulated by the ER model. Specifically, Figure 4.10

(a-b) show that penalized regression identified more true positives than PC, but also introduce more false positives, while `PenPC` algorithm significantly reduces the number of false positives, though some true positives are also removed. At the end, the `PenPC` has the lowest number of false positives plus false negatives, as measured by Hamming distance (HD) (Figure 4.10(c)). Figures 4.10(d-f) show that across various cutoff values of α, `PenPC` consistently has better performance than the PC algorithm. Finally, Figure 4.10(g) shows the ROC curves for the `PenPC` and the PC algorithms, which illustrate that `PenPC` has better sensitivity and specificity than the PC algorithm regardless of the cutoff α. Similar conclusions can be drawn for the simulation results of BA models.

4.6 Application

We evaluated the performance of the `PenPC` algorithm using a gene expression dataset of *S.cerevisiae* where both observational and interventional data are available [Hughes et al., 2000]. The complete data were downloaded from `http://hugheslab.ccbr.utoronto.ca/supplementary-data/rii/`. After data processing following Maathuis et al. [2010], we obtained the final data sets of the expression of 5,361 genes for 63 control experiments (observational data) and 234 single-gene deletion mutants (interventional data). More precisely, in each deletion mutant, expression of one gene is either knocked out or knocked down. Both data sets were standardized such that expression for each gene had mean 0 and standard deviation 1. Assume the dependence of the 5361 genes' expression can be modeled by a DAG, denoted by $\mathcal{G} = (V, E)$, where $V = \{1, ..., 5361\}$ and E contains all the edges in this DAG. The purpose of our study is to estimate the skeleton of this DAG using the observational data (n=63), and then to evaluate the accuracy of skeleton estimates by comparing the causal effects estimated from the skeleton and the interventional effects estimated from the interventional data. Intuitively, more accurate skeleton estimate leads to better consistency between causal effects estimates

and interventional effects.

The interventional effects are defined as follows. Let $c(i)$ be the vertex that was deleted in the i-th mutation strain, and let $\boldsymbol{c} = \{c(i), i = 1, ..., 234\}$. Let $\boldsymbol{A} = \{a_{ij}\}_{234\times5361}$ be the 234×5361 interventional data matrix, where a_{ij} is the (standardized) expression of the jth gene in the i-th mutation strain. Using the interventional data $\boldsymbol{A}$, we define the interventional effect of $c(i) \to j$ for $c(i) \neq j$ as

$$|a_{i,j} - mean(a_{-i,j})|/|a_{i,c(i)} - mean(a_{-i,c(i)})|, \tag{4.4}$$

where $mean(a_{-i,j})$ is the mean expression of gene j across all the conditions rather than the i-th mutation strain. We refer to $|a_{i,j} - mean(a_{-i,j})|$ as (absolute) expression change of gene j upon perturbation of gene $c(i)$, denoted by $\delta_{i,j}$. The interventional effect of gene $c(i)$ on gene j is defined by the ratio of $\delta_{i,j}$ over $\delta_{i,c(i)}$. Figure 5 (a-b) show the distributions of the standardized expression and $\log_{10}$ interventional effects.

We applied both the PC algorithm and the `PenPC` algorithm to construct skeleton using the observational data where $n = 63$ and $p = 5361$. Following Maathuis et al. [2010], the conditional independence test p-value cutoff α, was chosen as 0.01. Then the skeleton is extended to completed partially directed acyclic graph (CPDAG) following the approach described in section 5.6. Then we applied intervention-calculus when the DAG is absent (IDA) method [Maathuis et al., 2009] on the CPDAG to estimate causal effect .

Figure 4.18 displays the scatter plot of the $234 \times 5361 - 234$ estimated causal effects from PC and `PenPC`. We divided the genes into four regions based on whether the causal effect estimates from the PC algorithm or the `PenPC` algorithm is larger than 0.8. As shown in Figure 4.18(d), the genes in region R3, where both algorithms produce large casual effect estimates, have the largest interventional effect, followed by regions R2 (`PenPC` produces large causal effect but PC does not), R3 (PC produces large causal

effect but `PenPC` does not), and R1 (neither algorithms produces large causal effect estimates). This order of R3, R2, R4, and R1 implies that many stronger causal effects are captured by the `PenPC` algorithm but missed by the PC algorithm.

Assume the top $m\%$ of the interventional effects are true positives and all the other interventional effects are false positives. We can calculate, among the top q estimated causal effects, the number of false positives $\texttt{fp}(m, q)$ and the number of true positives $\texttt{tp}(m, q)$. Figure 4.19 (a) displays the partial ROC curves by plotting $\texttt{fp}(m = 10, q)$ versus $\texttt{tp}(m = 10, q)$ for q up to 5000. For the PC algorithm, the ROC curve is the same as the curve of Figure 1-a in Maathuis et al. [2010]. `PenPC` algorithm dominates PC algorithm in almost all regions except for the small regions with strong estimated causal effects.

In order to investigate the performances for various m values, we approximate the partial area under the ROC curve (pAUC) as

$$pAUC(m, q) = \int_0^q ROC(m, q')dq'$$

where $ROC(m, q')$ is the ROC curve from $\texttt{fp}(m, q')$ and $\texttt{tp}(m, q')$. Figure 4.19 (b) displays the partial AUC values according to m and it shows dominant performance of the `PenPC` algorithm over the PC-algorithm for all m values except for regions where m is very small. We notice that those top interventional effects often correspond to small denominator values in the interventional effect definition: $|a_{i,j} - mean(a_{-i,j})|/|a_{i,c(i)} - mean(a_{-i,c(i)})|$ (Figure 4.20). In other words, those top interventional effects are often from those experiments where the targeted genes are only moderately knocked down. Therefore, it is likely that the interventional effects were inflated.

4.7 Order independent `PenPC` algorithm

PC algorithm is order-dependent because its output depends on the order in which the variables are given. Colombo and Maathuis [2012] modified the PC algorithm so that its result is order-invariant, and they named their new algorithm as the PC-stable algorithm, which shows substantially improved performance than the PC algorithm [Colombo and Maathuis, 2012]. Motivated by the PC-stable algorithm, we systematically investigate the order dependency of the `PenPC` algorithm. The order dependence in the `PenPC` algorithm occurs in two places : (1) Step 1 of the `PenPC` which estimates the Gaussian Graphical Model (GGM) (2) Step 2 of the `PenPC` which removes the false connections between parents sharing at least one common child. Next we study how much the order dependency in each step of the `PenPC` affects the final skeleton estimation, using the yeast gene expression data presented in the application section.

Figure 4.21 displays the variability from 51 random permutations of the ordering of the gene expression variables. For each permutation, we estimated a GGM using the step 1 of the `PenPC`. Among all the pairs of the 5361 variables, 99.7% are perfectly stable, i.e., they form edges or gaps across all 51 permutations. Among those perfectly stable pairs, 7377 form edges (the completely blue columns in Figure 4.21(a)). For an unstable pair of variables, which may form an edge or a gap across the permutations, we define $e_{ij}^k = 1$ if there is an edge between X_i and X_j in the kth permutation and $e_{ij}^k = 0$ otherwise. We define an instability measure

$$r_{ij}(1 - r_{ij})$$

where $r_{ij} = \sum_{k=1}^{K} I(e_{ij}^k = 1)/K$ for variable pair (X_i, X_j) across K permutations. Figure 4.21(b) displays the density curve of the total number of edges of the 51 estimated GGMs and the red point indicates the number of edges when we use the original order.

Figure 4.21(c) displays the density curve of instability values for unstable variable pairs forming edges or gaps across permutations. Although the highest peak is around instability 0.025, significant amount of the unstable edges are near the maximum value 0.25. The order dependency comes from coordinate descent algorithm because it partially optimize the objective function w.r.t. each one of the coefficients.

We further examine the order dependency of step 2 of the `PenPC` algorithm. We fix the GGM by estimating it from the original order. Therefore the observed order-dependency is only due to step 2 of the `PenPC` algorithm. Figure 4.22 show the results from 51 permutations of variable ordering for step 2 of `PenPC`. Comparing the instability measures for step 1 and step 2 of the `PenPC`, we conclude that step 2 is the main source of the order dependency.

To solve those two sources of the order-dependency, we introduce some modifications of the `PenPC` algorithm. For the penalized regression step, we order the covariates by their (absolute) correlations with the response. For conditional independence testing step, we use simple modification following Colombo and Maathuis [2012]. The algorithm is shown in Figure 4.23. The modified parts are highlighted in blue. We refer to the modified `PenPC` algorithm as order-independent `PenPC` algorithm.

In simulation studies, we compared the order-independent `PenPC` algorithm and the PC-stable algorithm. We followed the same data generation procedure as for ER model in the simulation section. Figure 4.25 shows the estimation performance of order independent `PenPC` algorithm and PC-stable algorithm. The PC-stable algorithm for n=50 shows similar results to Figure 4 of Colombo and Maathuis [2012]. Figure 4.25(a) shows that the number of edges detected by `PenPC` increases slower less than that of the PC-stable algorithm as α increases. The Hamming distance in the Figure 4.25(b) indicates that the PC-stable algorithm adds significant amount of false positives and false negatives as sample size increases especially for bigger α. The performance of

PenPC is more stable and it outperforms the PC-stable algorithm for larger sample sizes. We define the true discovery rate (TDR) as the proportion of edges in the estimated skeleton that are also present in the true skeleton. For samples size larger than 200, PenPC shows consistently better TDR than PC-stable algorithm for all α's in Figure 4.25(c). We evaluate the estimation accuracy of CPDAG by structural Hamming distance (SHD), which counts the minimum number of edge insertions, deletions, or flips that are needed in order to transform the estimated graph into the true one [Colombo and Maathuis, 2012]. Since a CPDAG is estimated by applying a set of deterministic rules to a skeleton, the estimation accuracy of a CPDAG is directly affected by the estimation accuracy of the corresponding skeleton. Therefore, as expected, for larger sample size, order-independent PenPC outperforms PC-stable algorithm in terms of estimation accuracy of CPDAG (Figure 4.25(d)).

4.8 Conclusions

We propose a two-step approach, the PenPC algorithm, to estimate skeletons of high dimensional DAGs. After estimating GGM in the first step, the skeleton estimation problem boils down to finding co-parent relationships in the second step. We show that the PenPC algorithm is asymptotically consistent for the skeleton of a high dimensional DAG. For fixed graphs, the number of vertices p_n could be exponential scale of the sample size n. The results could be extended to random graphs. We considered two commonly used random graph models and discussed in detail the conditions under which the consistency properties hold. The simulation studies and real data analysis show that the network skeletons estimated by PenPC are substantially more accurate than those estimated by the PC algorithm. We implemented our method in an R package PEN. In PenPC, the most demanding part of the computation is to estimate GGM using p separate penalized regressions. For example, our real data analysis where

p=5,361 and n=63 was performed on an 2.93 GHz Intel processor 12M L3 cache (model X5670) and 48GB RAM running on Linux using 64bit R2.12.2. The step 1 in `PenPC` algorithm took 30 seconds for one penalized regression using log-penalty with 100×10 2-dimensional tuning parameter search so that it is about $5,361 \times 30 = 160,830$ seconds in total to find neighborhoods for all vertices. However, the p separate penalized regression is possible to be performed under parallel computing. The modified PC-algorithm (the step 2 of the `PenPC` algorithm) is computationally much more efficient than the PC-algorithm. In the real data analysis, we started from GGM with 16,006 edges, and it took 342 seconds to run the modified PC-algorithm using function `skeletonPEN` in the `PEN` package with $\alpha = 0.01$. In contrast, the PC-algorithm took 6,719 seconds using function `skeleton` in `R/pcalg` package with $\alpha = 0.01$. Therefore `PenPC` algorithm has computational advantage if one needs to evaluate the results across a large number of αs. Furthermore, we notice that the computation time for PC algorithm increases rapidly as sample size or significance level for partial correlation testings increase. For example, for $p = 5361$ and $\alpha = 0.05$, the computation times are about 2, 4, 11, 22, 54, and 81 hours for sample sizes n=50, 100, 200, 300, 400 and 500 respectively. In contrast, the computational time for penalized regressions are almost invariant across these sample sizes.

4.9 Tables and figures

Table 4.1: Simulation Setting

p	n	p_E (ER)	e (BA)
11	100	0.2	1,2
100	30	0.02, 0.03, 0.04, 0.05	1,2
1000	300	0.002, 0.005, 0.01	1,2

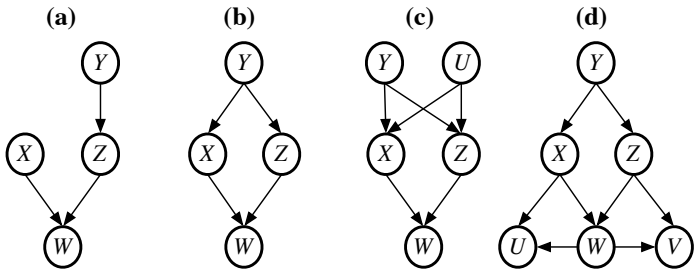

Figure 4.1: Four DAGs where X and Z are not connected in the skeleton, but are connected in the corresponding GGMs.

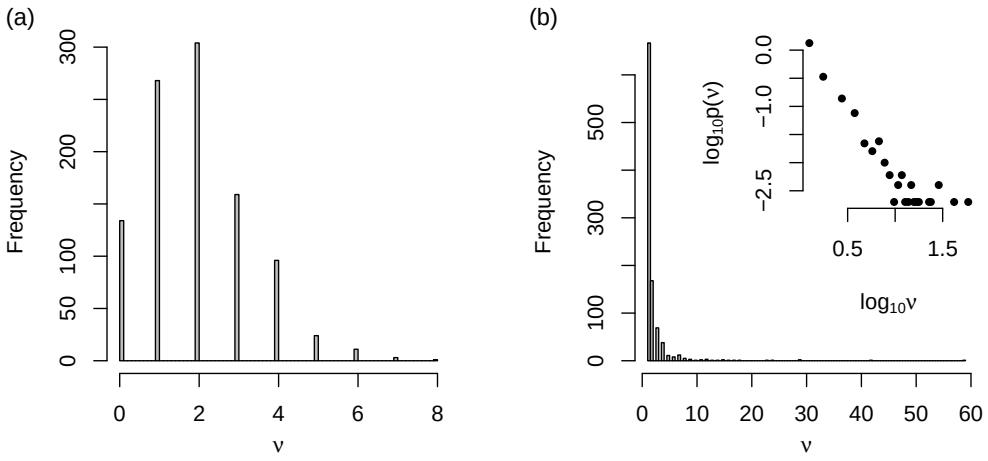

Figure 4.2: Histograms of the degree ν. (a) ER model with $p = 1000$ and $p_E = 2/p$. (b) BA model with $p = 1000$ and $e = 1$ and the $\log_{10}$ scale density of $\log_{10} \nu$ in its subplot.

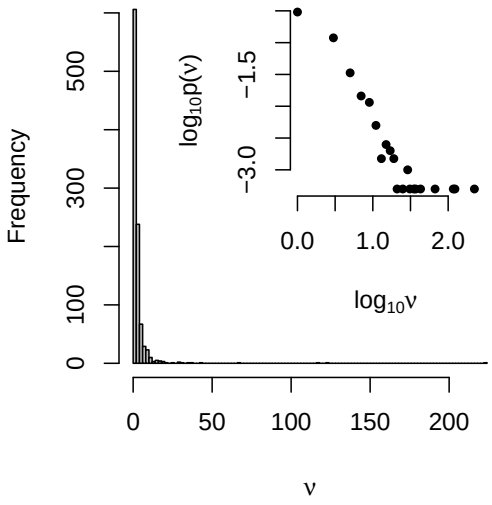

Figure 4.3: Histograms of the degree ν under BA model with $p = 1000$ and $e = 2$ and the $\log_{10}$ scale density of $\log_{10} \nu$ in its subplot.

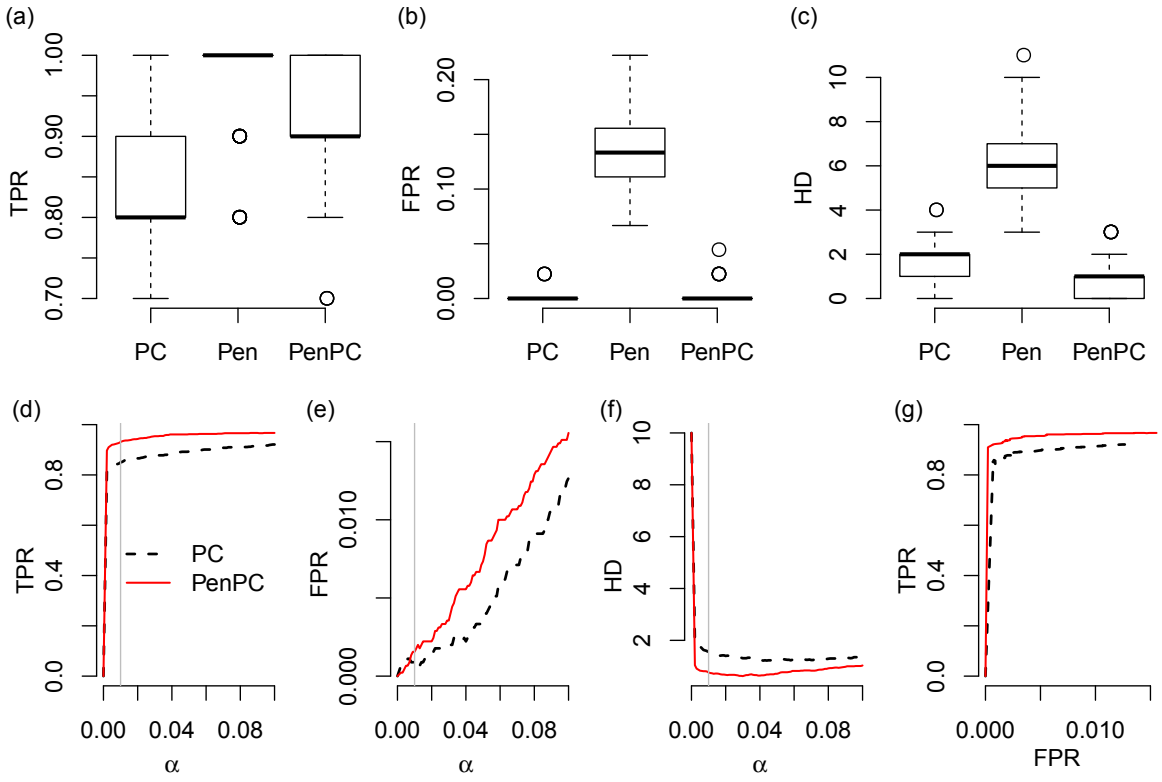

Figure 4.4: Performance of ER model ($p = 11, n = 100, p_E = 0.2$). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

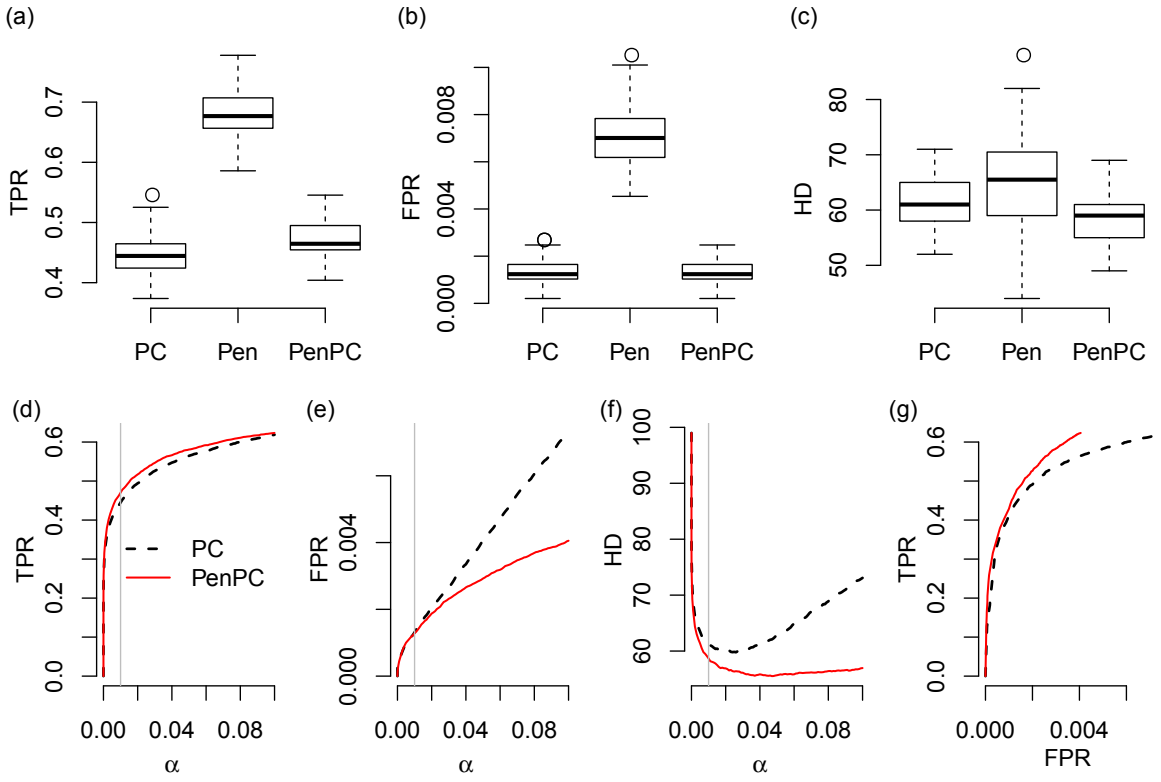

Figure 4.5: Performance of ER model ($p = 100, n = 30, p_E = 0.02$). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

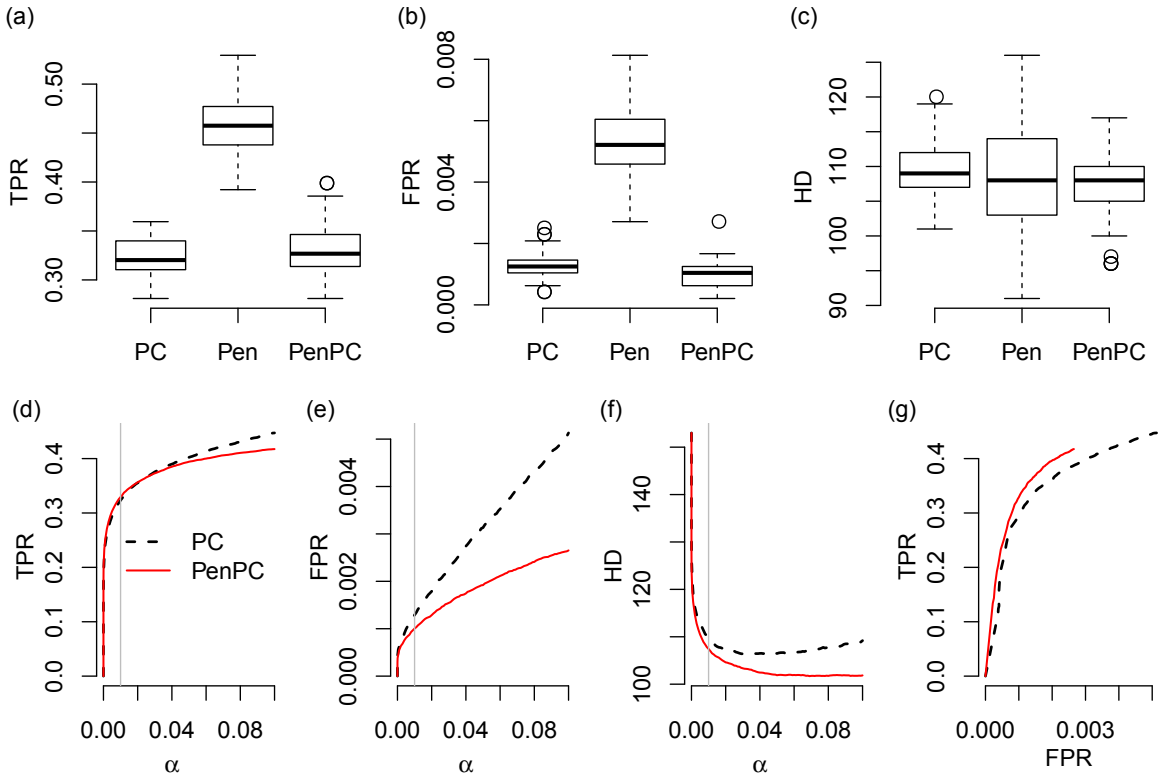

Figure 4.6: Performance of ER model ($p = 100, n = 30, p_E = 0.03$). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

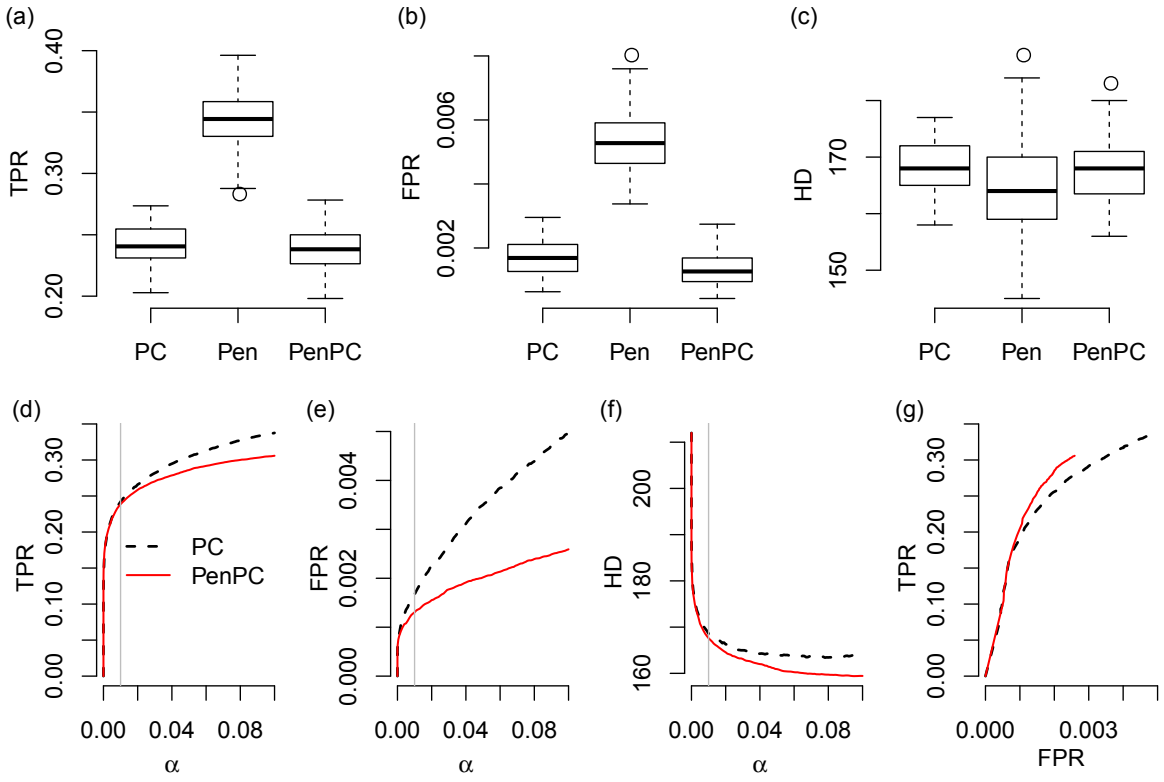

Figure 4.7: Performance of ER model ($p = 100, n = 30, p_E = 0.04$). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

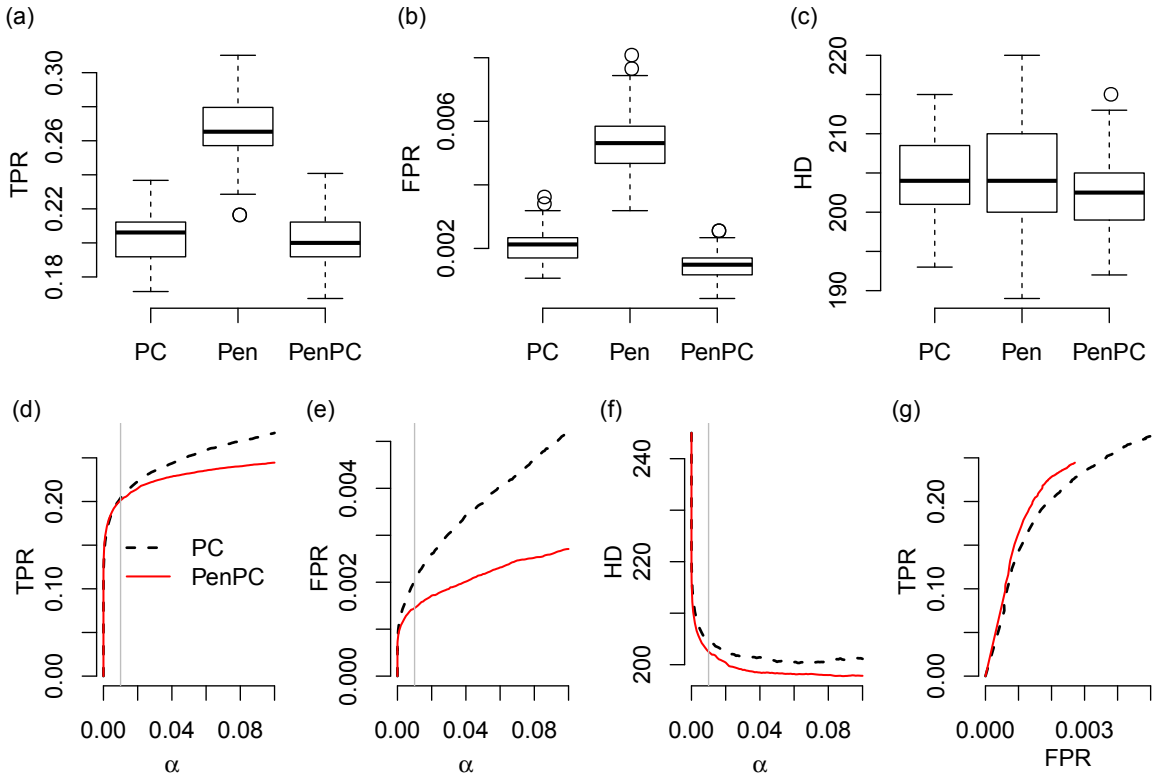

Figure 4.8: Performance of ER model ($p = 100, n = 30, p_E = 0.05$). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

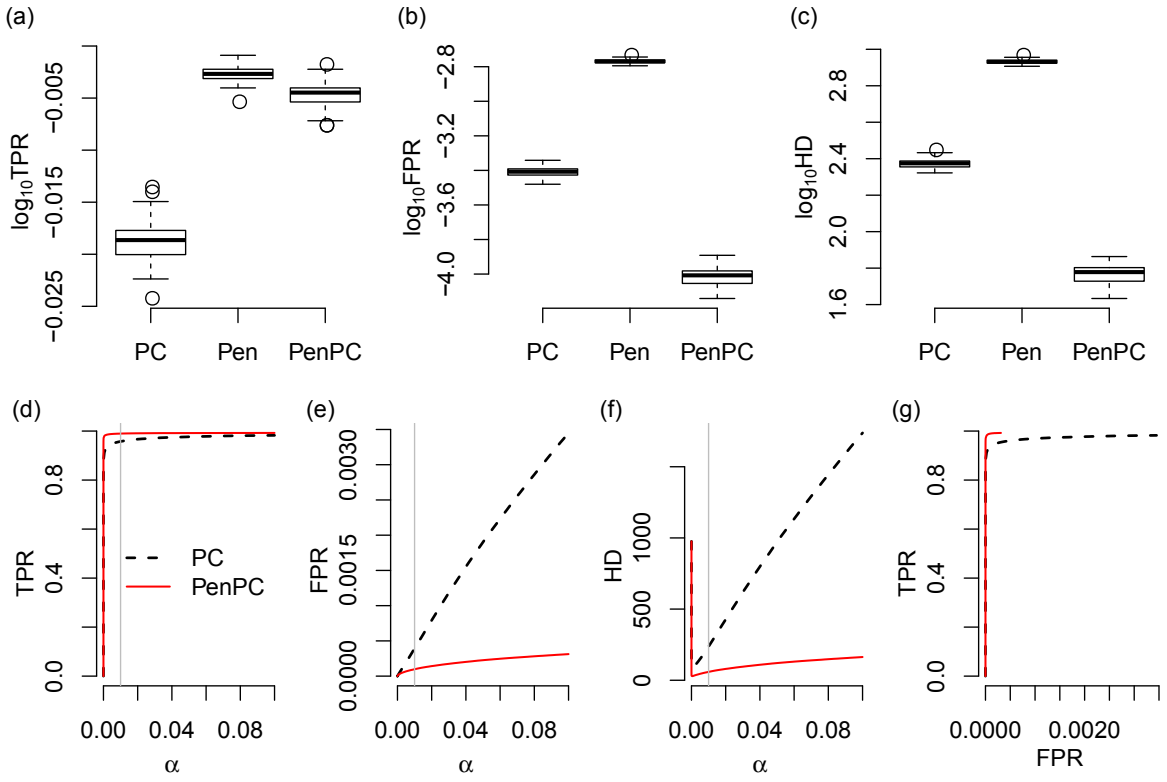

Figure 4.9: Performance of ER model ($p = 1000, n = 300, p_E = 0.002$). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

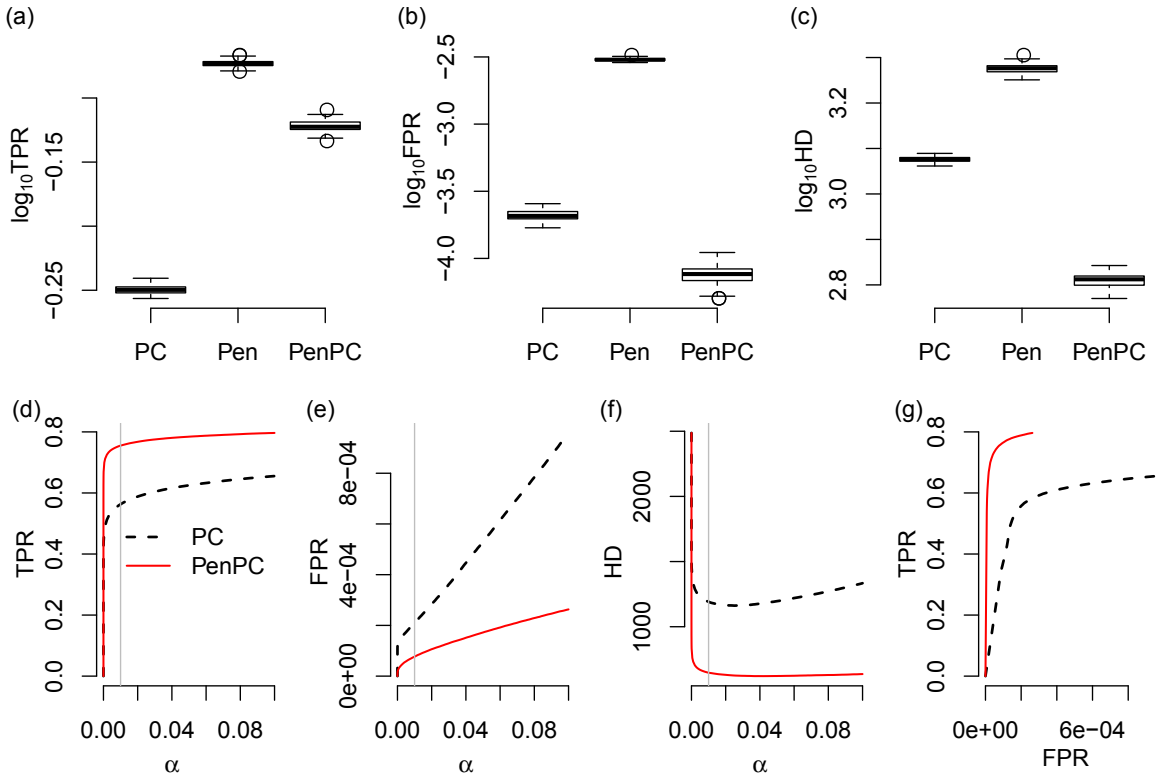

Figure 4.10: Performance of ER model ($p = 1000, n = 300, p_E = 0.005$). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

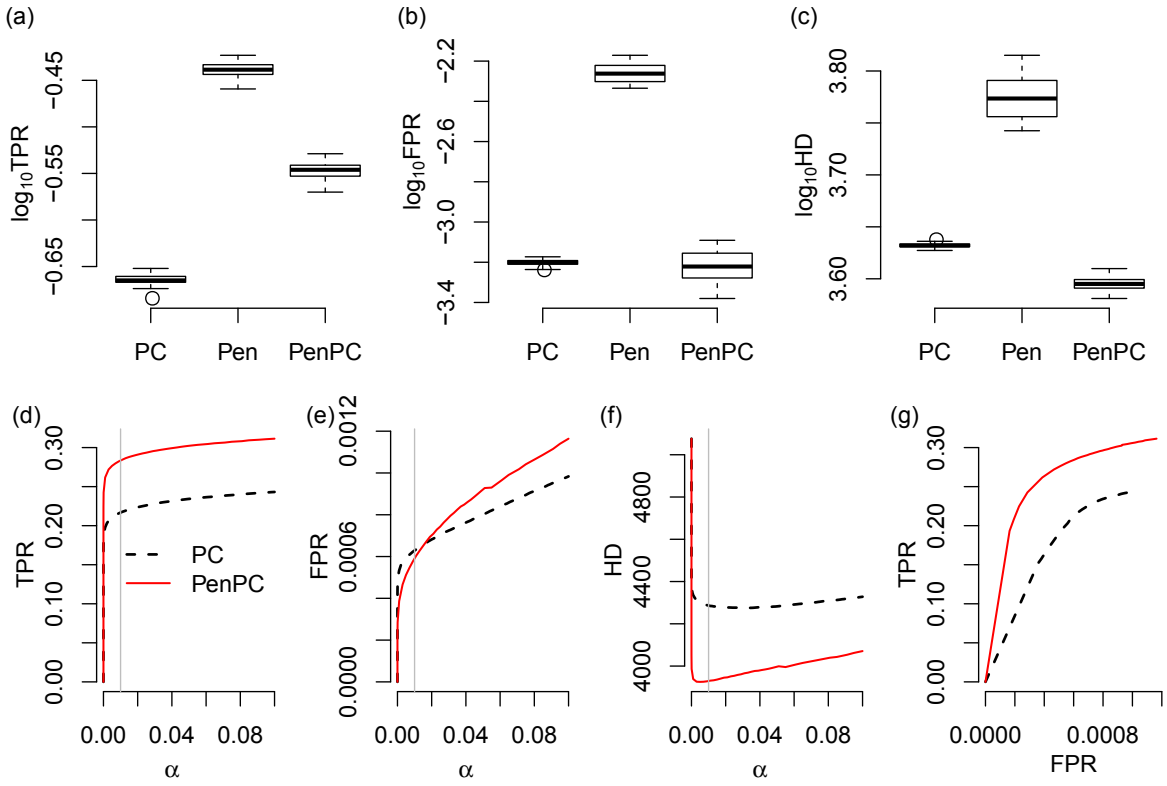

Figure 4.11: Performance of ER model ($p = 1000, n = 300, p_E = 0.01$). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

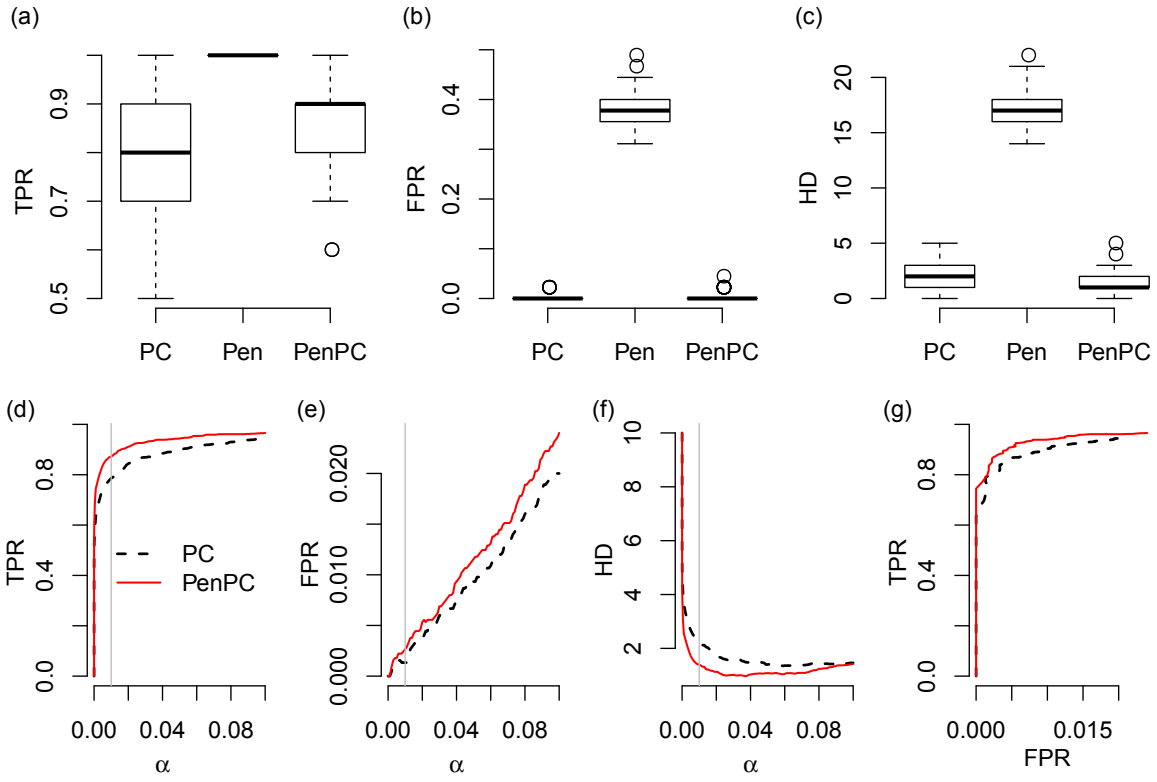

Figure 4.12: Performance of BA model (p=11,n=100,e=1). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

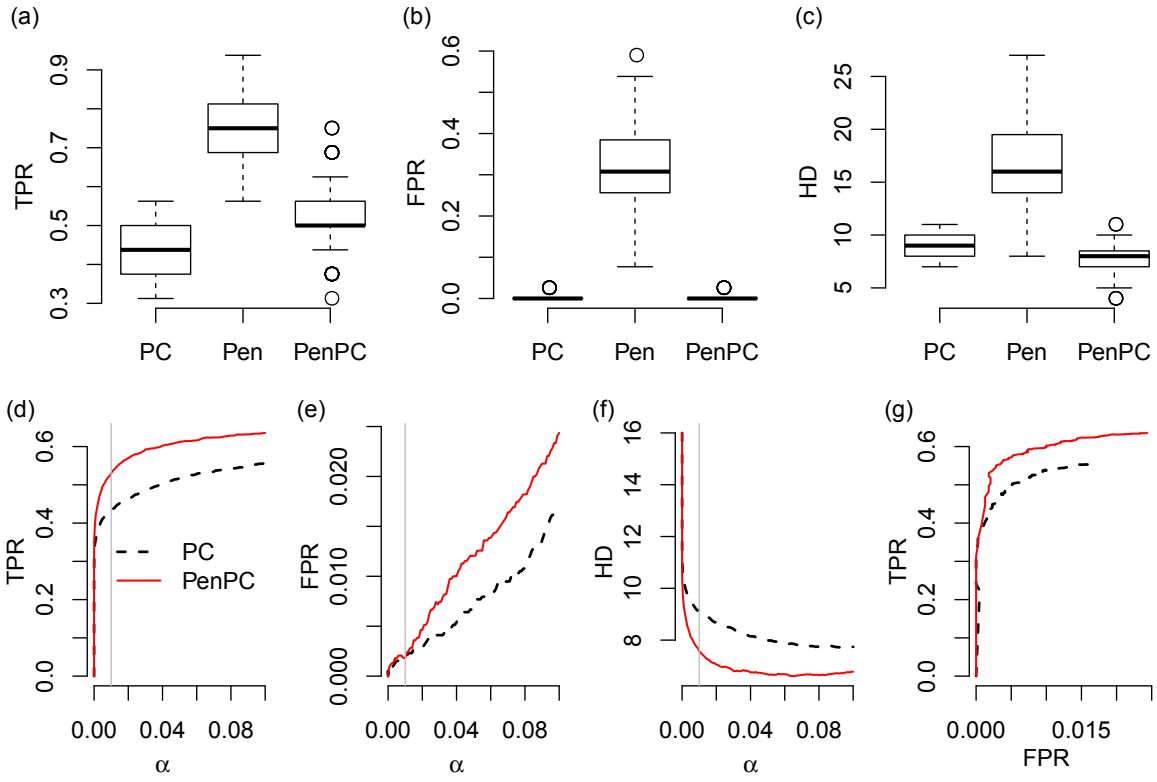

Figure 4.13: Performance of BA model (p=11,n=100,e=2). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

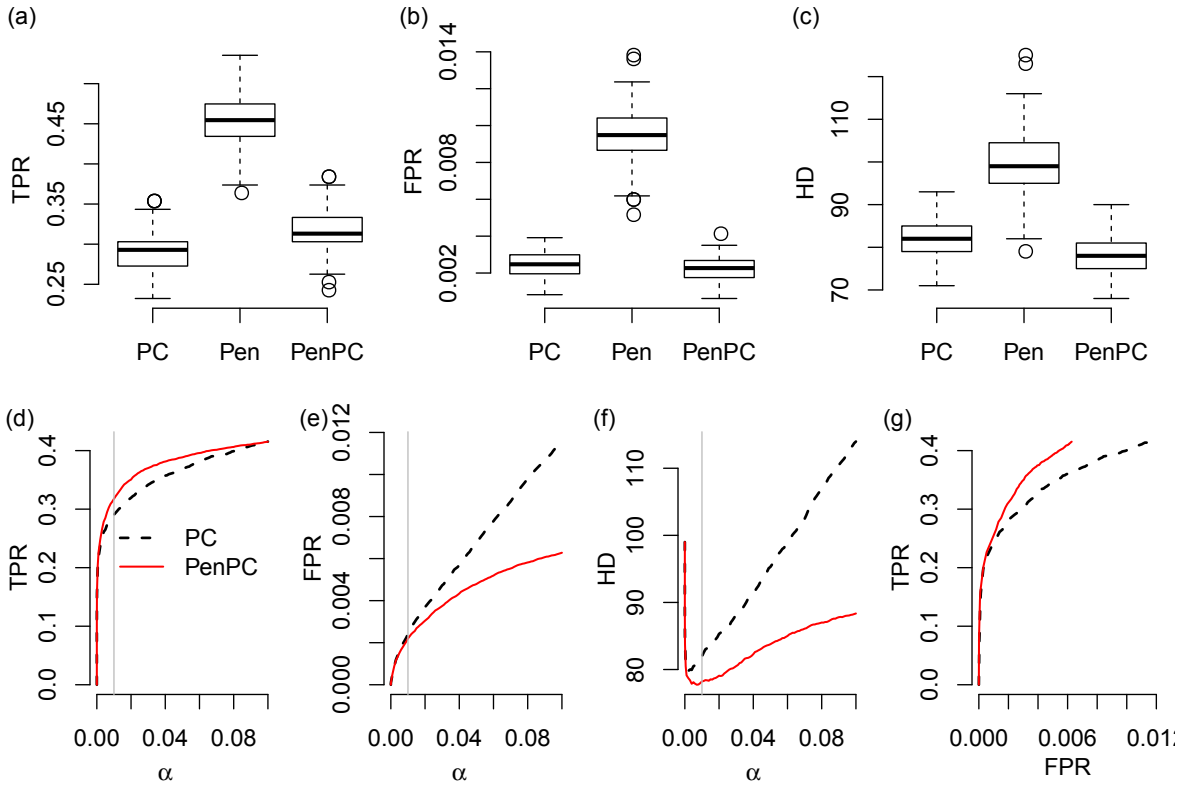

Figure 4.14: Performance of BA model (p=100,n=30,e=1). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

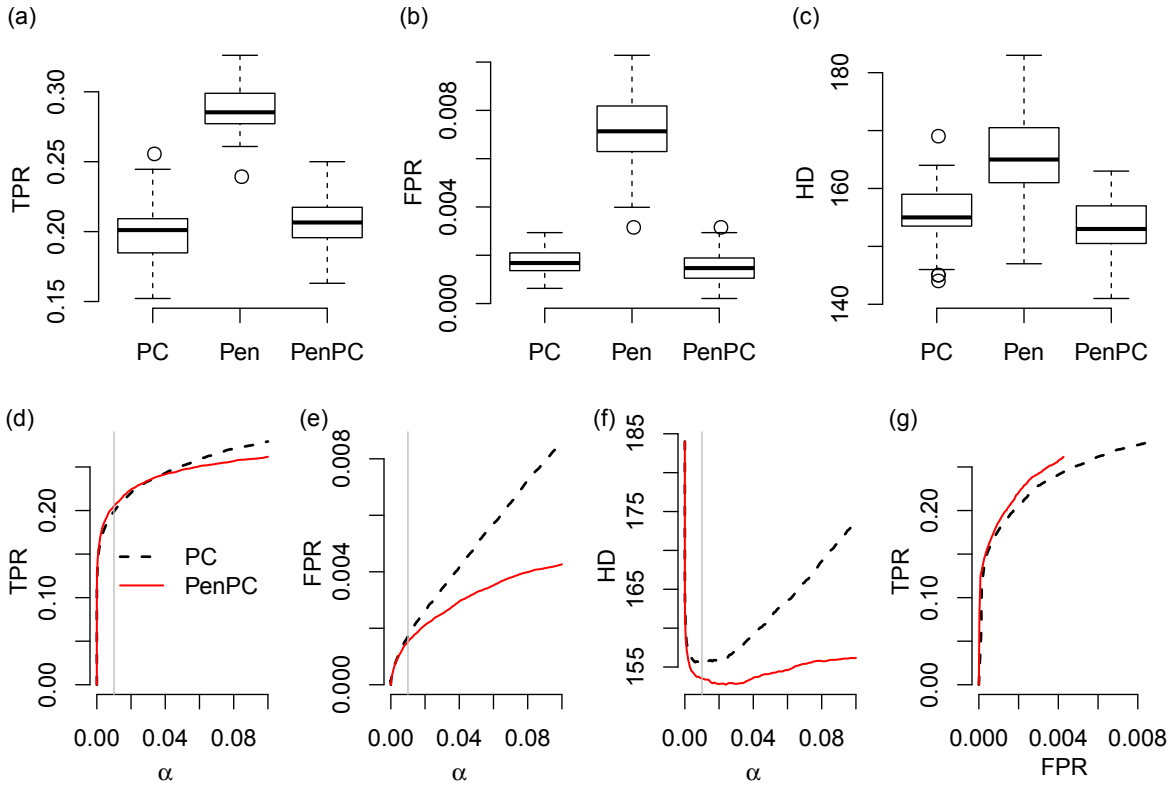

Figure 4.15: Performance of BA model (p=100,n=30,e=2). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

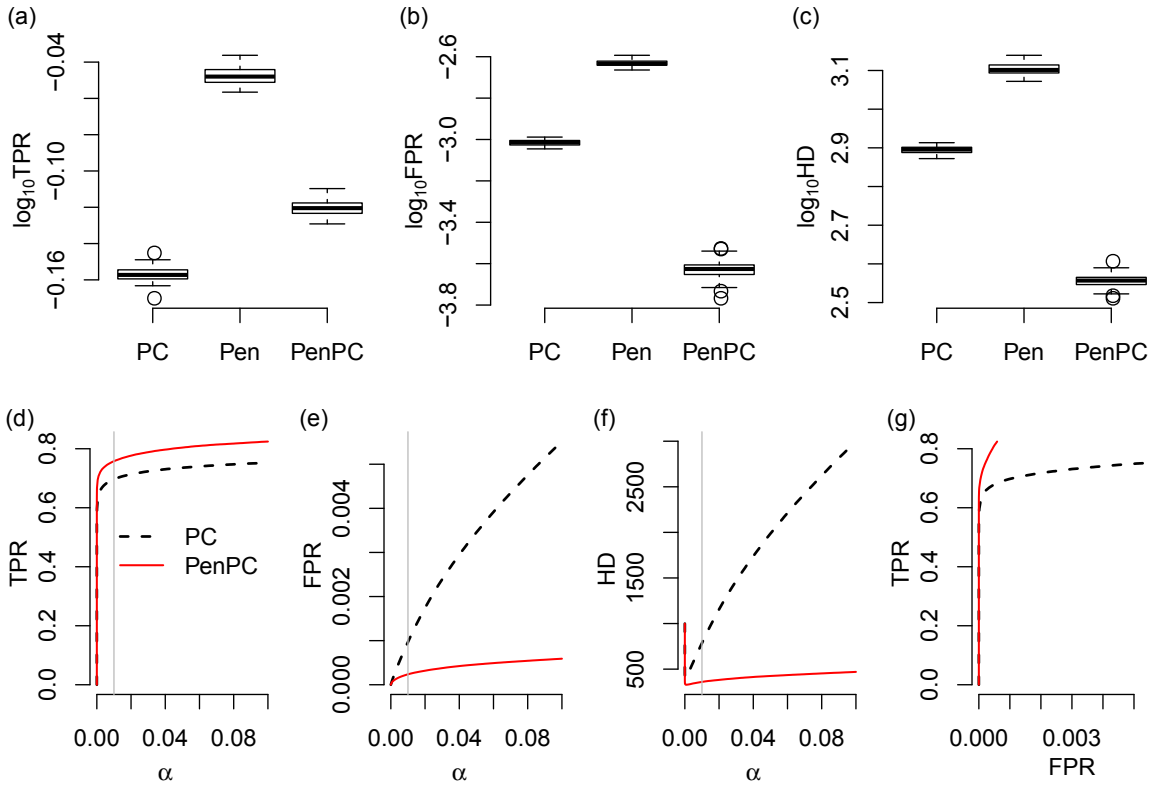

Figure 4.16: Performance of BA model (p=1000,n=300,e=1). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

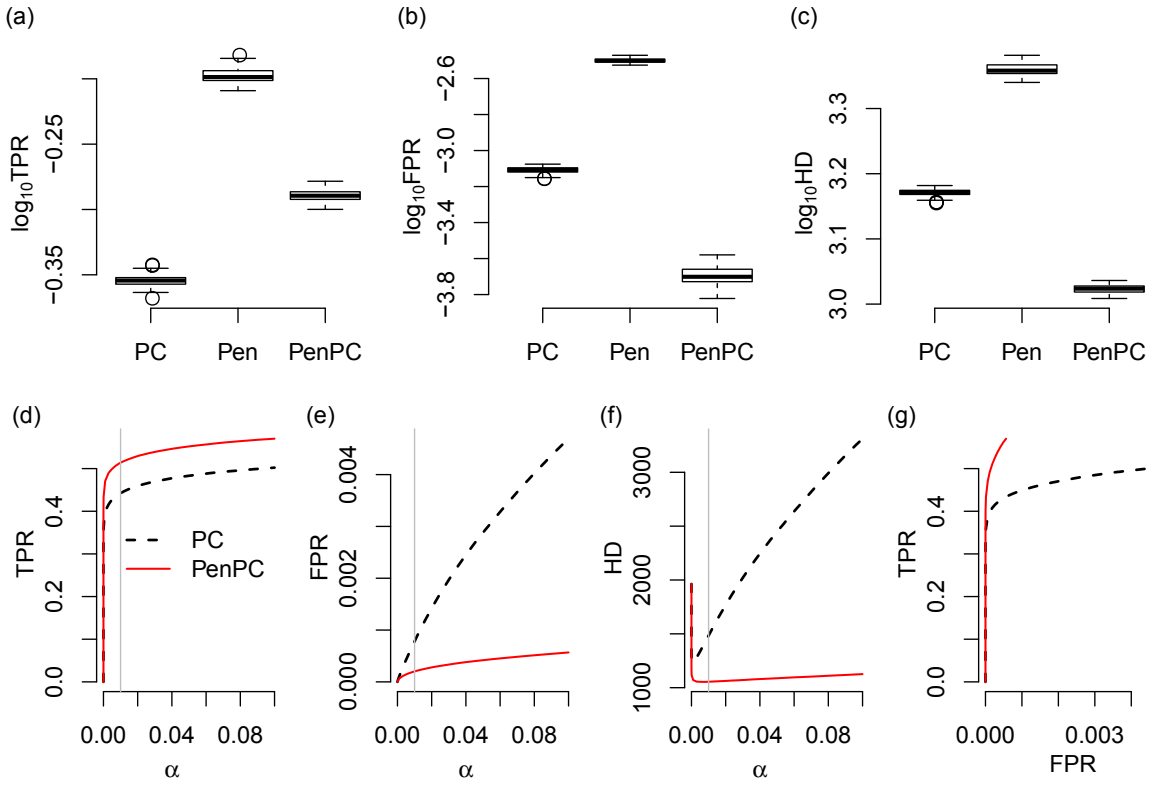

Figure 4.17: Performance of BA model (p=1000,n=300,e=2). The upper panels are box plots (in $\log_{10}$ scale) of true positive rate (TPR) (a), false positive rate (FPR) (b) and hamming distance (HD) (c) from 100 replications at $\alpha = 0.01$. The lower panels are average true positive rate (d), false positive rate (e), and Hamming distance (f) from 100 replications when the tuning parameter α is changed from 0 to 0.1 (the grey vertical line are at $\alpha = 0.01$). ROC curves are shown in panel (g).

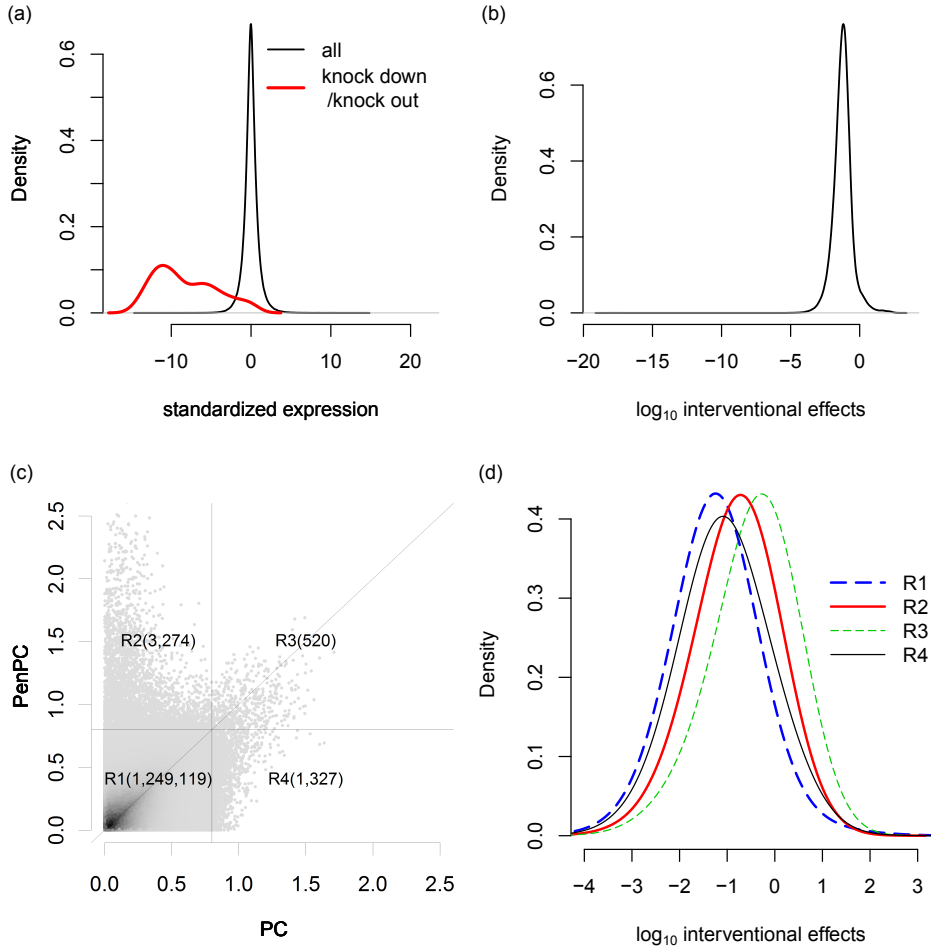

Figure 4.18: (a) The distribution of standardized gene expression of all the genes on all conditions (grey filled boxes) and standardized gene expression when a gene is knock down/knock out (black line boxes). (b) The density of $\log_{10}$ interventional effects. (c) The estimated causal effects from PC and `PenPC` algorithms, where regions R1-R4 are separated by horizontal/vertical lines at 0.8. (d) The distribution of $\log_{10}$ interventional effects according to regions in (c).

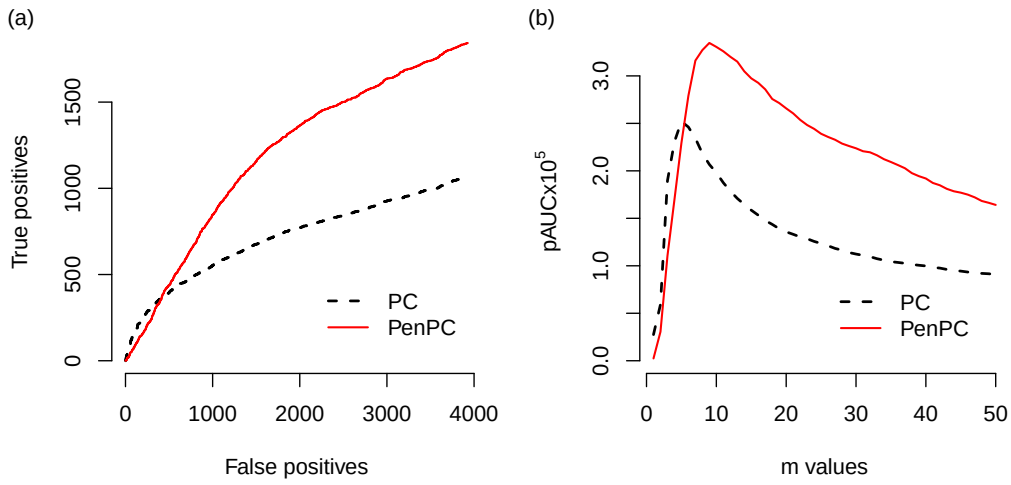

Figure 4.19: Performance of causal effects prediction. (a) The ROC (receiver operating characteristic) curves of the PC and `PenPC` algorithms, assuming the top m=10% of interventional effects are true positives. (b) The procedure (a) is repeated for m from 1 to 50 and the partial area under the ROC curve (pAUC) is plotted versus m values.

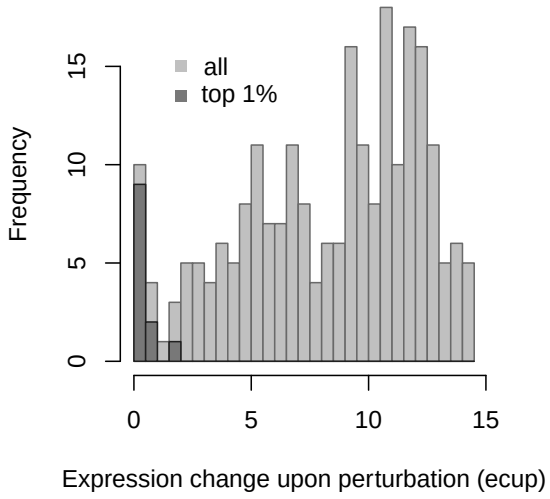

Figure 4.20: The distribution of "expression change upon perturbation" for all knock down/knock out genes (light grey) and those producing top 1% of the interventional effects

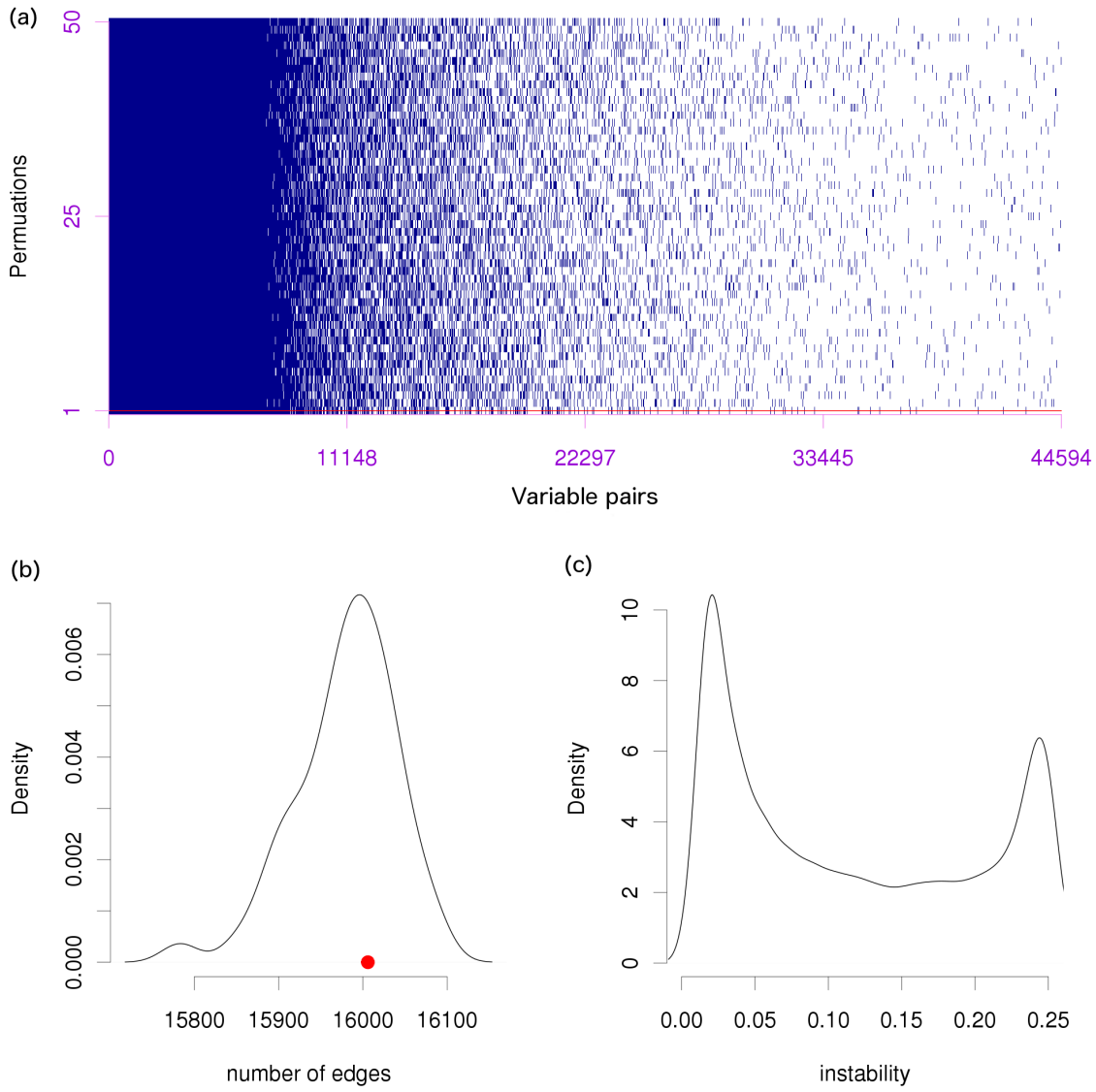

Figure 4.21: (a) Edge occurrence (indicated by dark blue) in the estimated GGMs for 50 random permutations of variable orders, as well as the original order (shown as the first permutation). The variable pairs along the x-axis are ordered by their frequencies of being connected (by length 1 chain) across 51 permutations (from 51 to 1) and the variable pairs that are not connected in any permutation are excluded. (b) The density curve of the total number of edges in the estimated GGMs from 51 different variable orders (black line) and the number of edges in the GGM with the original order (red point). (c) The density curve of instability values for unstable variable pairs

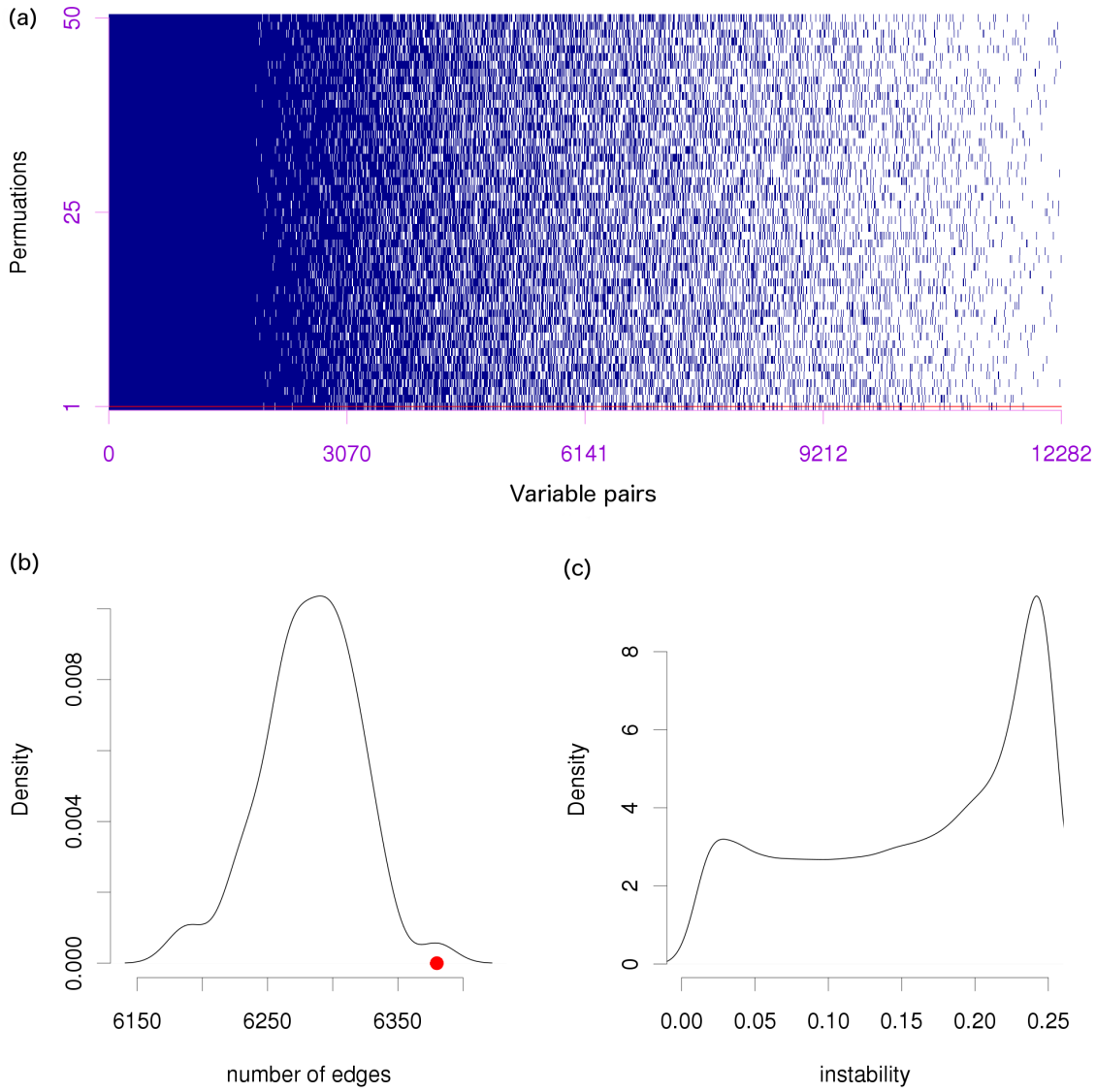

Figure 4.22: (a) Edge occurrence (indicated by dark blue) in the estimated skeletons with $\alpha = 0.01$ for 50 random permutations of variable orders, as well as the original order (shown as the first permutation). Here the step 2 of the `PenPC` algorithm is performed from the same GGM, which is estimated using the original ordering. The variable pairs along the x-axis are ordered by the frequencies of being connected (by length 1 chain) across 51 permutations (from 51 to 1) and the variable pairs that are not connected in any permutation are excluded. (b) The density curve of total number of edges in the estimated skeletons from 51 different variable orders (black line) and the number of edges in the skeletons with the original order (red point). (c) The density curve of instability values for unstable variable pairs.

Input: GGM $\mathcal{C}_{\mathcal{G}}$
Output: Skeleton $\mathcal{G}^u = (V, E^u)$ and separation set $S(i,j)$ for edges $(i,j) \notin E^u$ but $(i,j) \in F_{\mathcal{G}}$

1. **Set** l=-1 and $\mathbf{C} = \mathcal{C}_{\mathcal{G}}$ $(F = F_{\mathcal{G}})$
2. **For** all $(i,j) \in F$,
   2.1 if X_i and X_j are marginally independent, then delete (i,j) from F
3. **Repeat:** l=l+1
   3.1 $\tilde{\mathbf{C}} = \mathbf{C}$
   3.2 **Repeat:** Select an edge $(i,j) \in F$ such that $|\mathbf{\Gamma}(\tilde{\mathbf{C}})_{i,j}| \geq l$
      3.1.1 **Repeat:** Select $\Gamma \subseteq \mathbf{\Gamma}(\tilde{\mathbf{C}})_{i,j}$ with $|\Gamma| = l$
         3.1.1.1 Set $\mathcal{K} = [\mathtt{adj}(i, \tilde{\mathbf{C}}) \cup \mathtt{adj}(j, \tilde{\mathbf{C}})] \backslash [\Gamma \bigcup \{i,j\}]$
         3.1.1.2 If X_i and X_j are conditionally independent given $\{X_k : k \in \mathcal{K}\}$, then
         - Delete (i,j) from F
         - Save $\mathcal{K}$ in separation set for i and j, $S(i,j)$
      3.1.2 **Until:** The edge (i,j) is deleted from F or all $|\Gamma| = l$ have been chosen
   3.3 **Until:** All edges $(i,j) \in F$ with $|\mathbf{\Gamma}(\tilde{\mathbf{C}})_{i,j}| \geq l$ are tested for all conditioning set $\Gamma \subseteq \mathbf{\Gamma}(\tilde{\mathbf{C}})_{i,j}$ with $|\Gamma| = l$
4. **Until:** for each $(i,j) \in F$, $|\mathbf{\Gamma}(\tilde{\mathbf{C}})_{i,j}| < l$

Figure 4.23: Order-independent `PenPC` algorithm

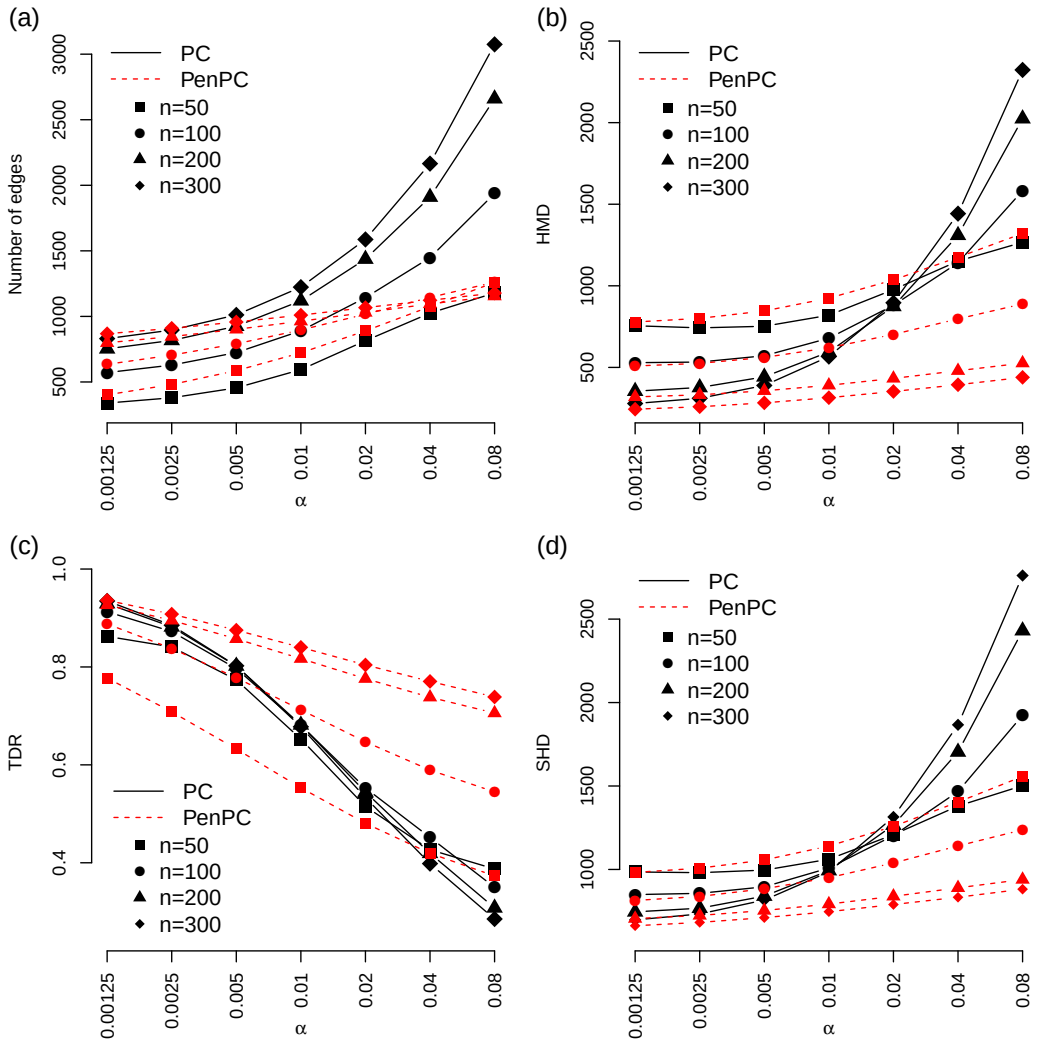

Figure 4.24: Estimation performance of order independent `PenPC` versus PC-stable algorithm for different values of α and sample size n in the ER model with $p = 1000$ and $p_E = 0.002$. The results are average from 100 randomly generated graphs. (a) Number of edges of skeleton estimates. (b) Hamming distance. (c) True discovery rate. (d) Structural Hamming distance.

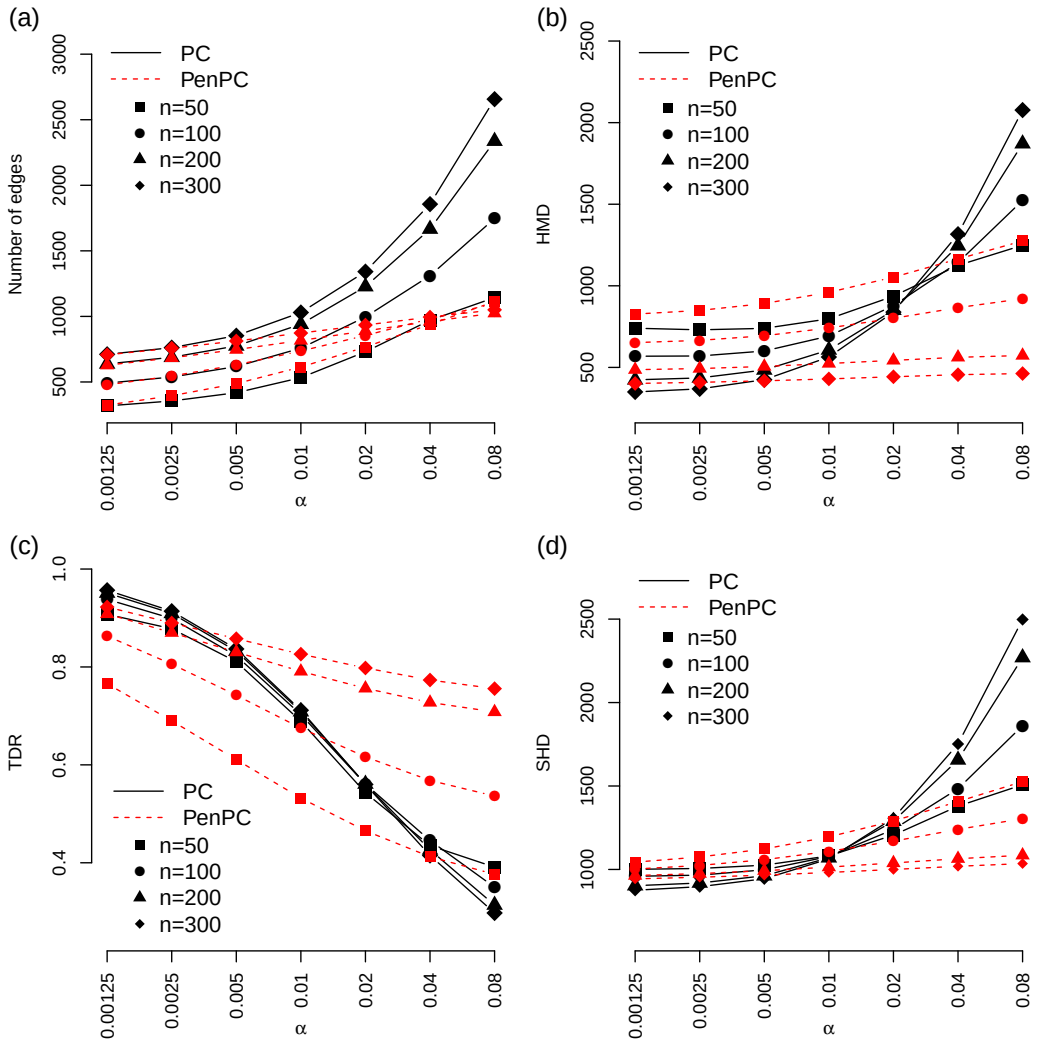

Figure 4.25: Estimation performance of order independent `PenPC` versus PC-stable algorithm for different values of α and sample size n in the BA model with $p = 1000$ and $p_E = 0.002$. The results are average from 100 randomly generated graphs. (a) Number of edges of skeleton estimates. (b) Hamming distance. (c) True discovery rate. (d) Structural Hamming distance.

Chapter 5

Estimation of High Dimensional Directed Acyclic Graphs with Surrogate Experiments

5.1 Introduction

Causal relationships among a set of random variables are represented by arrows in a directed acyclic graph (DAG), where vertices denote the variables and directed edges between some pairs of vertices constitute no directed cycle. Consider a DAG $\mathcal{G} = (V, E)$ whose vertices $V = \{Y_1, \ldots, Y_p\}$ correspond to random variables $Y_1, \ldots, Y_p$ and the set of edges E generate no directed cycle. We assume that $Y = (Y_1, \ldots, Y_p)^{\mathrm{T}} \in \mathbb{R}^p$ follows $\mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma}_Y)$ with density function $f_{\mathbf{\Sigma}_Y}(\cdot)$. The Markov properties determined by the DAG $\mathcal{G}$ admit *recursive factorization* of the joint probability density functions of the variables

$$f_{\mathbf{\Sigma}_Y}(y_1, \ldots, y_p) = \prod_{i=1}^{p} f_{\mathbf{\Sigma}_Y}(y_i | y_{\mathtt{pa}_i}), \tag{5.1}$$

where $\mathtt{pa}_i$ indicates the parents of vertex $Y_i \in V$. Because several different DAGs may constitute the same set of conditional independence restrictions among the set of variables (the same factorization), the collection of all possible DAGs having the same statistical model is called a *Markov equivalence* class. Larger Markov equivalence class produces more uncertainty in causal relations among a set of variables.

We are particularly interested in gene regulatory networks, where a set of vertices stand for genes that encode functional agents of the cell such as proteins and

edges depict causal relationships between sources and targets for gene activities [Vignes et al., 2011]. *Mendelian Randomization*, i.e., the randomization of the alleles passing to daughter cells during meiosis, provides a setting that is analogous to a randomized experiment and admits causal inferences on gene expressions [Li et al., 2006]. The early example of Katan [2004] where Mendelian randomization is used to infer causal relations between phenotypes has been described in several review papers [Smith and Ebrahim, 2003; Smith, 2007; Sheehan et al., 2008]. Consider assessing the causal effect of serum cholesterol levels (X) on cancer (Y) (Figure 5.1). It is clear that the causal effect is unidentifiable from the factorizations of the joint density under $X \to Y$ or $X \leftarrow Y$, and it is infeasible in reality to control the serum cholesterol level by intervention. Katan [2004] used *apolipoprotein E* (*ApoE*) gene to identify that the relation between low cholesterol levels and cancer is causal. Specifically, the *ApoE* gene can be considered as the direct cause of the serum cholesterol level because it is known to lower the cholesterol level. If cholesterol level was a causal factor for cancer, individuals with the genotype of *ApoE* gene associated with lower cholesterol should be expected to have higher cancer risk (Figure 5.1(a)). However if reverse causation is true and no confounding factor exists, no association would be expected between *ApoE* genotype and cancer (Figure 5.1(b)). The former situation was observed which justifies the causal relation that low cholesterol levels cause cancer.

In this chapter, we consider to use gene expression Quantitative Trait Loci (eQTLs), which are the genetic variants that affect gene expression, to orient the network skeleton of gene expression data. In eQTL studies, two types of data are collected from the same set of subjects: gene expression and genotypes of DNA polymorphisms [Kruglyak and Storey, 2009]. The Mendelian randomization of eQTL genotypes can be considered as *surrogate interventional experiment* [Pearl, 2000; Bareinboim and Pearl, 2012]. The utility of QTL or eQTL data in network analysis have been systematically assessed by

previous works [Li et al., 2006; Zhu et al., 2007; Neto et al., 2008; Chen et al., 2007; Millstein et al., 2009; Cai et al., 2013b].

5.2 Use QTL or eQTL data to infer phenotype networks

This section reviews approaches exploiting genetic variation to infer phenotype networks. Several methods have been developed for a special scenario, QTL-phenotype-phenotype triads which are sets constituted by a QTL and two phenotypes mapping to that QTL [Li et al., 2010]. Since a QTL can affect a trait directly, or indirectly through another intermediary trait, likelihood based conditional independence tests are widely used to distinguish causal, reactive, and independent relationships among the QTL-phenotype-phenotype triads [Schadt et al., 2005]. Motivated by this approach, Kulp and Jagalur [2006] allow for the interaction between genotype and phenotype to identify Quantitative Trait Genes (QTG), and Sun et al. [2007] detect relationships among QTG, transcription factor activity, and gene expression, and Chen et al. [2008] uncovers the components of coexpression networks that respond to variations in DNA. Zhu et al. [2007, 2008] construct Bayesian network by incorporating the eQTL data in determination of direction priors.

Several recent publications pursue joint inference of network and genetic architecture of the correlated phenotypes. Covariate-adjusted sparse precision matrices or conditional Gaussian graphical models (GGM) were proposed by Yin and Li [2011, 2013]; Cai et al. [2013a]. The *QTLnet* method [Neto et al., 2010] uses Bayesian model averaging with a modified Metropolis-Hastings algorithm to estimate causal phenotype networks. Hageman et al. [2011] propose a Bayesian methods where causal relationships between variables are described with hierarchical regression models including QTLs.

Another approach is to employ structural equation models (SEM) that permit both cyclic and acyclic graphs. Li et al. [2006] used score based model selection. Logsdon

and Mezey [2010] fits p (the number of phenotypes) separate adaptive lasso regressions for neighborhoods selection adjusting for the pre-selected eQTLs and then transform the resulting undirected graph into a DAG or a directed cyclic graph (DCG) by *recovery* theorem they proposed. Cai et al. [2013b] extended the work of Logsdon and Mezey [2010] by providing the adaptive Lasso the initial parameter estimates from the penalized regression using Lasso penalty.

Despite the success of previous works, DAG estimation using high dimensional gene expression data remains a very challenging question. We propose a new that to estimate DAGs in two steps. We first estimate the network skeleton of a DAG (an undirected graph after removing all edge directions of a DAG) using gene expression data and then orient the network skeleton using eQTL data. The PC algorithm and related methods are among the most popular methods for DAG skeleton estimation [Kalisch and Bühlmann, 2007; Maathuis et al., 2009; Colombo and Maathuis, 2012]. We have developed a new method named PenPC, which combines penalized regression with the PC algorithm and shows substantial advantage than the PC algorithm when the sample size is not too small. We will use the PC algorithm or the PenPC algorithm to construct DAG skeleton. To orient the skeleton using eQTL data, we use a model an averaging approach with some approximations to improve computational efficiency. In contrast to the most existing methods that require at least one eQTL per gene to construct DAG, our method allows a small proportion of genes having eQTL.

Gene expression abundance is traditionally measured by gene expression arrays. Recently, RNA-seq data are replacing microarray to be the standard platform. We have developed a statistical method that can identify eQTLs which have direct effects on the expression of a gene [Sun, 2012]. Using RNA-seq, two types of expression data are available. First, the expression of a gene can be estimated using the total number of sequence reads mapped to that gene, known as the total read count (TReC). We

assume properly normalized TReC data follow multivariate Gaussian distribution, and we construct gene expression network with the TReC data. Second, RNA-seq data also provide allele-specific gene expression (ASE) that is not available from microarray gene expression data. The combination of TReC and ASE from RNA-seq data can distinguish *cis*-eQTL and *trans*-eQTL [Sun, 2012; Sun and Hu, 2013]. *Cis*-eQTLs are DNA variations of a gene that directly influence transcript levels of that gene in an allele-specific manner. On the other hand, *trans*-eQTLs indirectly affect the expression of a gene by modifying the activity of the factors that regulate the gene, which leads to the same amount of expression changes for both alleles [Doss et al., 2005; Sun and Hu, 2013]. In our analysis, we only use *cis*-acting eQTL so that each eQTL is a direct cause of a gene.

5.3 Method

Recall that we seek to study the causal relations of p variables $Y_1, \ldots, Y_p$ by a DAG $\mathcal{G} = (V, E)$ with vertices $V = \{Y_1, \ldots, Y_p\}$. Let X be an additional set of variables so that they are direct causes of the variables $Y_1, \ldots, Y_p$ and they are subject to interventions. Let $\mathcal{I} \subseteq V$ be the set of vertices that are associated with at least one variable in X. For example in eQTL studies, $\mathcal{I}$ is the set of vertices in gene expression network with at least one eQTL. For any $Y_i \in \mathcal{I}$, $X^{(i)} = (X_1^{(i)}, \ldots, X_{q_i}^{(i)})^{\mathrm{T}} \in \mathbb{R}^{q_i}$ denotes the q_i variables which directly influence Y_i ($X_j^i \to Y_i$ for all $j = 1, \ldots q_i$). Let n be the sample size and denote the $n \times p$ observed data matrix of Y by $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_p)$. Following the notations in Pearl [2000]; Bareinboim and Pearl [2012], we denote the interventional values on variables $X^{(i)}$ by $\hat{\mathbf{x}}^{(i)} = (\hat{\mathbf{x}}_1^{(i)}, \ldots \hat{\mathbf{x}}_{q_i}^{(i)})$. The sufficient conditions for interventions on $X^{(1)}, \ldots, X^{(p)}$ to be surrogate experiments require that for each $i \in V$, $X^{(i)}$ has no direct effect of variables Y_j for $j \in V \setminus \{i\}$ [Pearl, 2000]. In this

study, we adopt this assumption, which can be justified by the fact that we use *cis*-acting eQTL. In this chapter we call the vertices in $V = \{Y_1, ..., Y_p\}$ as observational vertices and $X_j^{(i)}$ for all $j = 1, \ldots, q_i$ and all $i = 1, ..., p$ as interventional vertices.

5.3.1 Estimation of Markov equivalence class

Using observational data, we can estimate the distribution of Y. The DAG $\mathcal{G}$ is not identifiable from the distribution of Y, $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_Y)$ because several different DAGs may determine the same factorization in equation (5.1). As previously mentioned in Chapter 2, two DAGs are Markov equivalent if and only if they have the same skeleton and the same v-structure [Andersson et al., 1997]. The *skeleton* of a DAG $\mathcal{G}$ is obtained by replacing all directed edges to undirected edges. We denote the skeleton of $\mathcal{G}$ by $\mathcal{G}^u = (V, E^u)$ where $(Y_i, Y_j) \in E^u \Leftrightarrow (Y_i, Y_j) \in E$ or $(Y_j, Y_i) \in E$. A *v-structure* is an ordered triplet of vertices (Y_i, Y_j, Y_k) such that $\mathcal{G}$ contains the directed edges $(Y_i, Y_k) \in E$ and $(Y_j, Y_k) \in E$ and Y_i and Y_j are not adjacent in $\mathcal{G}$. In such a v-structure, the co-parents Y_i and Y_j share a common child Y_k which is called a *collision* vertex. All the DAGs that are Markov equivalent form a Markov equivalence class, which can be estimated from observation data.

Next we describe a few relevant concepts. The distribution of Y is *faithful* to $\mathcal{G}$ if and only if (i) for any vertex pair (Y_i, Y_j) in V, $(Y_i, Y_j) \in E^u$ if and only if Y_i and Y_j are dependent conditional on every subsets in $V \setminus \{Y_i, Y_j\}$ and (ii) in a *v-structure* $Y_i \rightarrow Y_k \leftarrow Y_j$, Y_i and Y_j are marginally independent or conditionally independent given the parents of Y_i and Y_j, but Y_i and Y_j are dependent given every set that contains Y_k or its descendants [Spirtes et al., 2000]. Given a Markov equivalence class, a directed edge is *compelled* if this edge exists in every DAG in the equivalence class, and it is *reversible* otherwise. The edges participating in v-structures of a DAG are compelled edges. A partially directed acyclic graph (PDAG) is a graph that contains

both directed and undirected edges with no directed cycle. A PDAG can used to represent a Markov equivalence class [Chickering, 2002]. The *completed* PDAG (CPDAG) corresponding to a Markov equivalence class is the PDAG consisting of directed edges for all compelled edge in the equivalence class, and undirected edges for all reversible edges in the equivalence class [Chickering, 2002].

The Inductive Causation (IC) algorithm aims at estimating the CPDAG of a DAG and the algorithm consists of three steps: (1) estimation of the skeleton $\mathcal{G}^u$ by a set of conditional independence tests, (2) v-structure identification, and (3) completion of the PDAG obtained from (1) and (2) [Pearl, 2009]. After estimating skeletons using conditional independence tests, the v-structures are simply determined for any triples such that (Y_i, Y_k, Y_j) with $(Y_i, Y_k) \in E^u$, $(Y_k, Y_j) \in E^u$ but $(Y_i, Y_j) \notin E^u$ by assigning a v-structure $Y_i \rightarrow Y_k \leftarrow Y_j$ if Y_k is not included in the conditioning set which make Y_i and Y_j independent. The completion in the step (3) of the IC algorithm is to maximally orient the undirected edges as possible with restriction of no directed cycle. This completion can be done by applying the rules suggested by Meek [1995a] and the resulting PDAG is shown to be a CPDAG.

The skeleton estimation procedure of the IC algorithm is the part to which statistical methods can contribute. Spirtes et al. [2000] describe various algorithms to estimate the skeleton. SGS (Spirtes-Glymour-Schein) algorithm starts from a complete undirected graph where any pair of vertices are connected then thins the graph by removing the edges such that $Y_i - Y_j$ is removed if Y_i and Y_j are conditionally independent given a subset $S \subseteq V \setminus \{Y_i, Y_j\}$. PC (Peter and Clark) algorithm thins the complete graph by sequentially removing edges with marginal independence, 1st order conditional independence, 2nd order conditional independence, and so on. The PC algorithm can be computationally much more efficient than the SGS algorithm, especially for sparse graphs. In the high dimensional and sparse setting, [Kalisch and Bühlmann,

2007] proved consistency of PC-algorithm when the number of vertices grows in polynomial order of the sample size. [Colombo and Maathuis, 2012] improved PC-algorithm by solving order dependency in the sense that the resulting skeleton depends on the variable ordering of the input data and the algorithm is called *PC-stable* algorithm. Another method, the IG (Independence Graph) algorithm first estimates undirected independence graph which includes all the edges in the skeleton plus edges between co-parents sharing a child, and then SGS algorithm is applied in each clique to exclude the co-parent relations. Our PenPC method is conceptually very similar to the IG algorithm. The differences include that we use penalized regression to obtain the initial undirected independence graph, and we use a modified PC-algorithm to remove the edges due to co-parent relations.

5.3.2 Edge orientation given surrogate experiments

Given the external variables $X^{(i)}$ for all $Y_i \in \mathcal{I}$, the recursive factorization of the joint probability density of the variables becomes

$$f_{\mathbf{\Sigma}_Y}(y_1, \ldots, y_p) = \prod_{Y_i \in \mathcal{I}} f_{\mathbf{\Sigma}_Y}(y_i | y_{\mathtt{pa}_i}, \hat{\mathbf{x}}^{(i)}) \prod_{Y_i \in V \setminus \mathcal{I}} f_{\mathbf{\Sigma}_Y}(y_i | y_{\mathtt{pa}_i}) \tag{5.2}$$

where $\hat{\mathbf{x}}^{(i)}$ includes q_i fixed intervention values. Now we consider to orient the skeleton $\mathcal{G}^u$. Given any undirected edge in $\mathcal{G}^u$, $Y_i - Y_j$, the hypothesis of interest is

$$\text{Model 1: } Y_i \to Y_j \text{ vs. Model 2: } Y_j \to Y_i. \tag{5.3}$$

Denote $\mathbf{D}$ as the data including $\mathbf{Y}$ and $\hat{\mathbf{x}}^{(i)}$ for all $i = 1, \ldots, p$. Our interest is to

calculate the posterior probability of $(Y_i, Y_j) \in E$ for any Y_i and Y_j in V

$$\begin{aligned} \mathbb{P}\left(Y_i \to Y_j | \mathbf{D}\right) &= \frac{\sum_{t=1}^{T} I\left(Y_i \to Y_j \in E_t | G_t, \mathbf{D}\right) \mathbb{P}(G_t|\mathbf{D})}{\sum_{t=1}^{T} \mathbb{P}(G_t|\mathbf{D})} \quad (5.4) \\ &= \frac{\sum_{t=1}^{T} I\left(Y_i \to Y_j \in E_t | G_t, \mathbf{D}\right) \mathbb{P}(\mathbf{D}|G_t)}{\sum_{t=1}^{T} \mathbb{P}(\mathbf{D}|G_t)}, \quad (5.5) \end{aligned}$$

where $G_t = (V, E_t)$ for $t = 1, \ldots T$ are all possible DAGs given the DAG skeleton. The equation (5.5) is obtained by assuming all the G_t's for all $t = 1, \ldots, T$ have the same prior probability, which is reasonable since they all have the same number of edges. Let $\mathtt{pa}_k^t$ be the set of parents of Y_k in the graph G_t. We define the local likelihood of Y_k given its parents and corresponding interventional variables as

$$L_n(Y_k | Y_{\mathtt{pa}_k^t}, \hat{\mathbf{x}}^{(k)}) = \exp\left\{ -\frac{n \log \sigma_k^2}{2} - \frac{\|\mathbf{y}_k - \mathbf{y}_{\mathtt{pa}_k^t} \boldsymbol{\beta}_k - \hat{\mathbf{x}}^{(k)} \boldsymbol{\alpha}_k\|^2}{2\sigma_k^2} \right\}, \quad (5.6)$$

where $\mathbf{y}_k$ is an $n \times 1$ vector including observations of variable Y_k, $\mathbf{y}_{\mathtt{pa}_k^t}$ is an $n \times |\mathtt{pa}_k^t|$ matrix including the observations of the parents of Y_k in G_t, $\hat{\mathbf{x}}^{(k)}$ is an $n \times q_k$ matrix including the observations correspond to the interventional variables of Y_k, $\boldsymbol{\beta}_k$ is a $|\mathtt{pa}_k^t| \times 1$ coefficient vector and $\boldsymbol{\alpha}_k$ is a $q_k \times 1$ coefficient vector. Then by the Markov property, the edge direction posterior in equation (5.5) becomes

$$\mathbb{P}\left(Y_i \to Y_j | \mathbf{D}\right) = \frac{\sum_{t=1}^{T} I(Y_i \to Y_j \in E_t) \prod_{k=1}^{p} L_n(Y_k | Y_{\mathtt{pa}_k^t}, \hat{\mathbf{x}}^{(k)})}{\prod_{k=1}^{p} L_n(Y_k | Y_{\mathtt{pa}_k^t}, \hat{\mathbf{x}}^{(k)})}, \quad (5.7)$$

where $I(\cdot)$ is an indicator function. If $T = 2^N$ where N is the number of undirected edges in $\mathcal{G}^u$ is small, we can calculate the direction posterior probabilities in equation (5.7) from all possible G_t's.

The summation across all possible T DAGs in equation (5.7) is not computationally feasible for a network skeleton with a large number of edges. Therefore, instead of

evaluating all the possible DAGs within a module, we evaluate a large number of high-likelihood DAGs, which are identified by the following approach. If two genes are connected by an undirected edge and at least one of these two genes has an eQTL, we refer to such undirected edges as *starting edges*. We identify high-likelihood DAGs in two steps: (1) randomly assign the directions of all the starting edges based on their posterior probabilities; and (2) iteratively update the direction of each edge by assigning the direction with higher likelihood and acyclic constraints. The posterior probability in step (1) is easy to calculate, since the relevant graph is simply $X^{(i)} \rightarrow Y_i - Y_j \leftarrow X^{(j)}$. Step (2) is also computationally easy because we only consider the local graph $[X^{(i)} \bigcup \mathtt{pa}_i] \rightarrow Y_i - Y_j \leftarrow [X^{(j)} \bigcup \mathtt{pa}_j]$, where $\mathtt{pa}_i$ is the set of parents of Y_i identified by currently directed edges. We repeat steps (1) and (2) a number of times to obtain T high-likelihood DAGs. Finally we assign the directions of all the edges using equation (5.7) given a posterior probability cutoff. Choosing a posterior probability cutoff is a relatively easy task because posterior probability can be interpreted as a local False Discovery Rate (local FDR) and overall FDR is the summation of local FDRs Efron and Tibshirani [2002].

Next we describe the details of our algorithm.

(1) randomly assign the directions of all the starting edges based on their posterior probabilities. We assume the prior probabilities $\mathbb{P}(Y_i \rightarrow Y_j) = \mathbb{P}(Y_j \rightarrow Y_i) = 0.5$ for any starting edge $Y_i - Y_j$ in the skeleton, then

$$\begin{aligned} \mathbb{P}(Y_i \rightarrow Y_j|\mathbf{D}) &= \frac{L_n(Y_i \rightarrow Y_j|\mathbf{D})\mathbb{P}(Y_i \rightarrow Y_j)}{L_n(Y_i \rightarrow Y_j|\mathbf{D})\mathbb{P}(Y_i \rightarrow Y_j) + L_n(Y_j \rightarrow Y_i|\mathbf{D})\mathbb{P}(Y_j \rightarrow Y_i)} \\ &= \frac{L_n(Y_j|Y_i, \hat{\mathbf{x}}^{(j)})L_n(Y_i|\hat{\mathbf{x}}^{(i)}))}{L_n(Y_j|Y_i, \hat{\mathbf{x}}^{(j)})L_n(Y_i|\hat{\mathbf{x}}^{(i)}) + L_n(Y_i|Y_j, \hat{\mathbf{x}}^{(i)})L_n(Y_j|\hat{\mathbf{x}}^{(j)})}. \end{aligned} \tag{5.8}$$

By the definition of starting edges, at least one of X_i and X_j is not an empty set. For each starting edge, we randomly generate edge direction according to this posterior

probability with acyclic constraint. The resulting graph is called an *initial graph*.

(2) iteratively update the direction of each edge by assigning the direction with higher likelihood. Given the initial graph, we orient the edges as follows. For each edge between variables Y_i and Y_j, we still assume the prior probabilities $\mathbb{P}(Y_i \to Y_j) = \mathbb{P}(Y_j \to Y_i) = 0.5$. We abuse the notation a little bit to denote the current working graph as G_t so that G_t is being updated in each step of our algorithm. The posterior probability of $\mathbb{P}(Y_i \to Y_j|\mathbf{D}, G_t)$ is

$$\frac{L_n(Y_j|Y_i, Y_{\mathtt{pa}_j^t}, \hat{\mathbf{x}}^{(j)})L_n(Y_i|Y_{\mathtt{pa}_i^t}, \hat{\mathbf{x}}^{(i)}))}{L_n(Y_j|Y_i, Y_{\mathtt{pa}_j^t}, \hat{\mathbf{x}}^{(j)})L_n(Y_i|Y_{\mathtt{pa}_i^t}\hat{\mathbf{x}}^{(i)}) + L_n(Y_i|Y_{\mathtt{pa}_i^t}, Y_j, \hat{\mathbf{x}}^{(i)})L_n(Y_j|Y_{\mathtt{pa}_j^t}, \hat{\mathbf{x}}^{(j)})}. \tag{5.9}$$

Then we decide the direction of edge $Y_i - Y_j$ with simple posterior comparison such that $Y_i \to Y_j$ if $\mathbb{P}(Y_i \to Y_j|\mathbf{D}, G_t) > \mathbb{P}(Y_j \to Y_i|\mathbf{D}, G_t)$ and $Y_j \to Y_i$ otherwise. The remaining question is how to decide the orders to orient all the edges, which is described in the following algorithm.

1. Randomly select a starting edge from the initial graph. Suppose this edge is $Y_i \to Y_j$.
2. Set $\Psi = \{Y_i\}$ and Ψ_0 as an empty set, where the former are the vertices to be considered and the latter are the vertices that have been considered.
3. While (Ψ is not an empty set)

3.1. For all ($Y_k \in \Psi$)

3.2.1 Construct Ψ_k, the vertices that are connected to Y_k by undirected edges.

3.2.2 For all ($Y_l \in \Psi_k$)

(1) Choose a direction $Y_k \to Y_l$ vs. $Y_l \to Y_k$ based on posterior probability.

(2) If the chosen direction lead to a directed cycle, then remove the direction and keep an undirected edge $Y_k \to Y_l$.

3.2 Set $\Psi_0 = \Psi_0 \cup \Psi$.

3.3 Update $\Psi = \texttt{adj}(\Psi, G_t) \setminus \Psi_0$ where $\texttt{adj}(\Psi, G_t)$ is the set of vertices adjacent to any vertices in Ψ.

We generate graphs G_t for $t = 1, ..., T$ by repeating the above steps (1) and (2) T times.

5.4 Simulation

We simulate random DAGs following the ER model [Erdős and Rényi, 1960] with a connection probability p_E. Each DAG $\mathcal{G}$ has $p = 1000$ vertices for 1000 genes, among which q genes have eQTLs. To simply the model, we assume each of these q gene has one and only one eQTL. The the gene expression data are simulated as follows:

1. Construct a $p \times p$ matrix $A = (a_{ij})$ and a $p \times 1$ vector $\mathbf{b} = (b_j)$ where all elements are zero. The former are regression coefficients for gene-gene associations and the latter are regression coefficients of eQTL effect sizes.

2. With probability p_E, the lower triangle elements of A are replaced by independent realizations of random variable $2Z$, where $Z \sim \texttt{Exp}(5)$, and $\texttt{Exp}(5)$ denotes exponential distribution with parameter 5.

3. Simulate the genotype data $\boldsymbol{x}_i$ for $i = 1, \ldots, p$ by a multinomial distribution denoted by $\texttt{multinomial}(n, (1-p_m)^2, 2p_m(1-p_m), p_m^2)$, where p_m is a predetermined minor allele frequency, and $X_i = 0$, 1, or 2 with probabilities $(1-p_m)^2$, $2p_m(1-p_m)$, and p_m^2, respectively.

4. Randomly choose q elements of $\mathbf{b}$, and fill in values by realizations of random variable $2Z$, where $Z \sim \texttt{Exp}(2)$, and $\texttt{Exp}(2)$ denotes exponential distribution with parameter 2.

After obtaining a topological order of the set of vertices V using A, the column vectors of $n \times p$ data matrix $\boldsymbol{Y}$ are generated by the equations:

$$\boldsymbol{y}_j = \sum_{k \in \mathtt{pa}_j} a_{jk} \boldsymbol{y}_k + b_j \boldsymbol{x}_j + \epsilon_j$$

where $n = 300$, $\mathtt{pa}_j$ is determined by the simulated DAG, and $\epsilon_j \sim \mathcal{N}(0, I_n)$. We simulate $\boldsymbol{y}_j$ for $j = 1, ..., p$ sequentially and the n elements of $\boldsymbol{y}_j$ are scaled to have variance 1 after simulating each $\boldsymbol{y}_j$.

In the simulation studies, we compare our method, which we refer to as `siDAG`, with the QDG (QTL-directed dependency graph) method [Neto et al., 2008]. The QDG method requires every gene in the DAG has at least one eQTL. Therefore to evaluate QDG, we simulate eQTLs for all the $p = 1000$ genes. While to test our method `siDAG`, we assume that a subset of the p genes have eQTL. Specifically, we set q, the number of genes with eQTL, to be 10%, 50%, 80% or 100% of p. For both QDG and `siDAG`, we start from the same skeleton obtained by the PC-stable algorithm [Colombo and Maathuis, 2012] using $\alpha = 0.01$ as the significance level for each partial correlation testing. Figure 5.2 displays performances of skeleton estimation using PC-stable algorithm. We calculated numbers of edges, numbers of true/false positives, numbers of true/false negatives, and Hamming distance, which is the number of false positives plus the number of false negatives. To evaluate the resulting partially directed graphs from CPDAG (without using eQTL data), QDG and `siDAG`, we consider four measures among true positives in the estimated skeleton, number of undirected edges (`UTP`), number of correctly directed edges (`TTP`), number of incorrectly directed edges (`FTP`). Additionally, we define a distance masure

$$\mathtt{Distance} = 2 * \mathtt{FTP} + \mathtt{UTP}.$$

Figure 5.3 displays those four measures of resulting graphs obtained by PC, QDG and `siDAG` for $q = 100$, 500, 800 and 1000. The QDG algorithm has two steps. It first uses eQTLs to direct each edge in the skeleton and then iteratively refine the directions of the edges. In Figure 5.3, "eQTL only" indicates the results of using the first step of QDG or the first step of `siDAG` when all genes have eQTLs. The advantage of our method `siDAG` is that it is applicable when only a subset of the genes have eQTLs. In fact, the accuracy of DAG estimation is already much better than CPDAG (without using eQTLs) when only 10% of the genes have eQTLs.

5.5 Application

Our method is applied to a breast cancer study. We use RNA-seq data from tumor tissues for 550 female caucasian samples. After removing genes with low expression across most samples, we end with 18,827 genes. The expression of each gene within each sample is measured by total read count (TReC), and we use log transformed TReC, logTReC, in this study. A 550×18,827 residual data is obtained after taking out the linear effects of several important covariates, 75 percentile of logTReC (which captures read depth), plate, institution, genotype PCs and expression PCs. Our goal is to estimate a network among those 18,827 genes with the set of vertices $V = \{Y_1, \ldots, Y_{18827}\}$. The PC algorithm implemented in `pcalg` package of [Kalisch and Bühlmann, 2007] is computationally too intensive to estimate a skeleton for our data with $p = 18,827$ and $n = 550$. Instead, we start from Gaussian graphical model (GGM) using neighborhood selection [Meinshausen and Bühlmann, 2006] with log penalty. Using GGM in the PC algorithm is one of the possible modifications described in Spirtes et al. [2000]. To estimate a GGM, we have the following pre-screening process:

(1) Marginal correlations (r_0) are tested for all pairs of vertices. A edge is kept if the testing p-value is smaller than 10^{-5}. Denote the neighborhood of vertex i as

$\mathtt{ne}_0(i)$.

(2) For any $Y_i \in V$ and $Y_j \in \mathtt{ne}_0(i)$, calculate 1st-order partial correlations (r_1) $cor(Y_i, Y_j|Y_k)$ for all $Y_k \in \mathtt{ne}_0(i) \setminus \{Y_j\}$. If the maximum p-value is less than 0.01, set $\mathtt{ne}_1(i) = \mathtt{ne}_0(i)$. Otherwise, set $\mathtt{ne}_1(i) = \mathtt{ne}_0(i) \setminus \{Y_j\}$.

For any vertex $Y_i \in V$, we select the neighborhood denoted by $\mathtt{ne}(i)$ using a penalized regression with Y_i as response variable and the variables in $\mathtt{ne}_1(i)$, denoted by $Y_{\mathtt{ne}_1(i)}$, as covariates:

$$\hat{\boldsymbol{b}}_i = \underset{\boldsymbol{b}_i \in \mathbb{R}^{|\mathtt{ne}_1(i)|}}{\arg\min} \frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{Y}_{\mathtt{ne}_1(i)}\boldsymbol{b}_i)^{\mathrm{T}}(\boldsymbol{y}_i - \boldsymbol{Y}_{\mathtt{ne}_1(i)}\boldsymbol{b}_i) + n \sum_{j \in \mathtt{ne}_1(i)} p_{\boldsymbol{\theta}}(|b_{i,j}|), \tag{5.10}$$

where $\boldsymbol{y}_i$ is $n \times 1$ vector for n measurements of variable Y_i, $\boldsymbol{Y}_{\mathtt{ne}_1(i)}$ is $n \times |\mathtt{ne}_1(i)|$ matrix for n measurements of $\boldsymbol{Y}_{\mathtt{ne}_1(i)}$, $\boldsymbol{b}_i \in \mathbb{R}^{\mathtt{ne}_1(i)}$ is the coefficient vector with elements $\{b_{ij}\}_{j \in \mathtt{ne}_1(i)}$ and $p_{\boldsymbol{\theta}}(|b_{i,j}|)$ denotes a penalty function with tuning parameters $\boldsymbol{\theta}$. After p penalized regressions, we obtain a Gaussian graphical model estimates denoted by $\hat{\mathcal{G}}^m = (V, \hat{E}^m)$ and the edge set $\hat{E}^m$ is estimated by

$$(Y_i, Y_j) \in \hat{E}^m \Leftrightarrow Y_i \in \mathtt{ne}(j) \text{ and } Y_j \in \mathtt{ne}(i).$$

Denote $\mathtt{adj}(Y_i, G)$ as the set of adjacent vertices to Y_i in a graph G. We run PC-algorithm as follows:

(1) Set $G = (V, F)$ where $F = E^m$.

(2) Repeat the following from $k = 2$ until $k > \max_{i \in V} \mathtt{adj}(Y_i, G)$

- For any $Y_i \in V$ and any $Y_j \in \mathtt{adj}(Y_i, G)$, calculate k-order partial correlations (r_k), $cor(Y_i, Y_j|Y_{\mathcal{K}})$ for all $Y_{\mathcal{K}} \subseteq \mathtt{adj}(Y_i, G) \setminus \{Y_j\}$ and $|Y_{\mathcal{K}}| = k$. If the maximum p-value is greater than or equal to 0.01, remove Y_j from $\mathtt{adj}(Y_i, G)$.

- Update F with

$$(Y_i, Y_j) \in F \Leftrightarrow Y_i \in \texttt{adj}(Y_j, G) \text{ and } Y_j \in \texttt{adj}(Y_i, G).$$

Figure 5.4-(a) shows distribution of neighborhood sizes after pre-screening procedure. From the marginal correlation screening, the sizes range from 4 to 11,984 and are reduced to range from 1 to 1,689 after 1st-order partial correlation screening. The PC algorithm proceeds until 11th-order partial correlation testing. Figure 5.4-(b) displays the degree distribution for the estimated GGM and PC-algorithm after $k = 2, 3, 4$ and 11. From the GGM, the degree ranging from 3 to 23 and 48,891 edges are kept. From the PC algorithm, the 11-order partial correlation testing gives degree ranging from 3 to 11 and 39,111 edges which is 0.02% of the all possible edges. After $k = 11$, the PC-algorithm stop because there is no edge having degree greater than 11.

If the network skeleton form several disconnected component, we can run our `siDAG` algorithm for each component separately. However, the estimated skeleton constructs a huge connected component including 18,210 vertices. We consider *community structure* of the skeleton. Community structure is the gathering of vertices into groups such that there is a higher density of edges within groups than between them and *modularity* is a division of a network into communities [Clauset et al., 2004]. After the analysis of the community structure on the skeleton, we obtain 34 modules with more than two vertices. All the other vertices are combined into another module. Figure 5.5 shows the relationship between the number of vertices and the number of edges in each module.

The next step is to direct the edges in the skeleton estimates using *cis*-eQTLs for 6156 genes, where we keep the most significant *cis*-eQTL per gene. The *cis*-eQTLs are identified by eQTL mapping using both Total Read Count (TReC) and allele-specific expression (ASE) Sun [2012]. After applying our method, `siDAG` for each module

skeleton, we have 23,370 directed edges and 522 undirected edges within modules. The estimated FDR was 0.001. The example graphs around a few breast cancer related genes such as ERBB2, ESR1, and FGFR2 are shown in Figures 5.6, 5.7, and 5.8, respectively.

5.6 Conclusion

A DAG is used to represent causal relationships among a set of variables. We focus on estimating a DAG that represents a set of random variables following multivariate Gaussian distribution. Estimation of the DAG based on observational data is a challenging problem because the conditional independence relations implied by the distribution satisfying Markov property may represent several DAGs. We have developed a method to estimate the DAG when there is an additional set of variables, which are subject to interventions and they are direct causes of the variables in the DAG. Simulation studies demonstrate the satisfactory performances of our method. We apply our method to construct a regulatory network from high dimensional gene expression data where we use genotype data of DNA polymorphisms as surrogate interventional data.

5.7 Figures

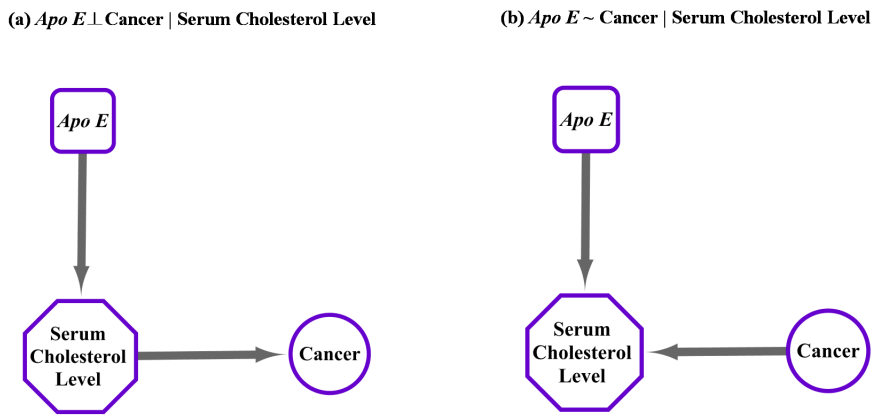

Figure 5.1: Example

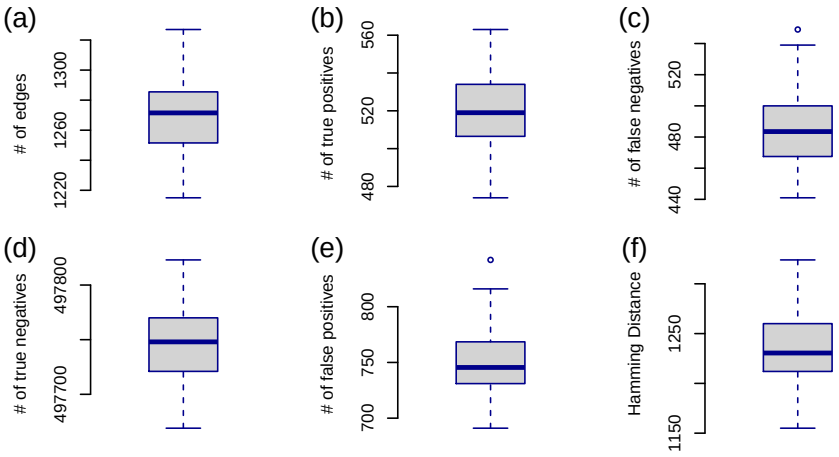

Figure 5.2: Performances of the estimated skeleton from PC-stable algorithm (p=1000, n=300, $p_m = 0.3$). Among 100 replications, 37 PDAGs (v-structures) were not extendable to a DAG. (a) Number of edges. (b) Number of true positives. (c) Number of false negatives. (d) Number of true negatives. (e) Number of false positives. (f) Hamming Distance.

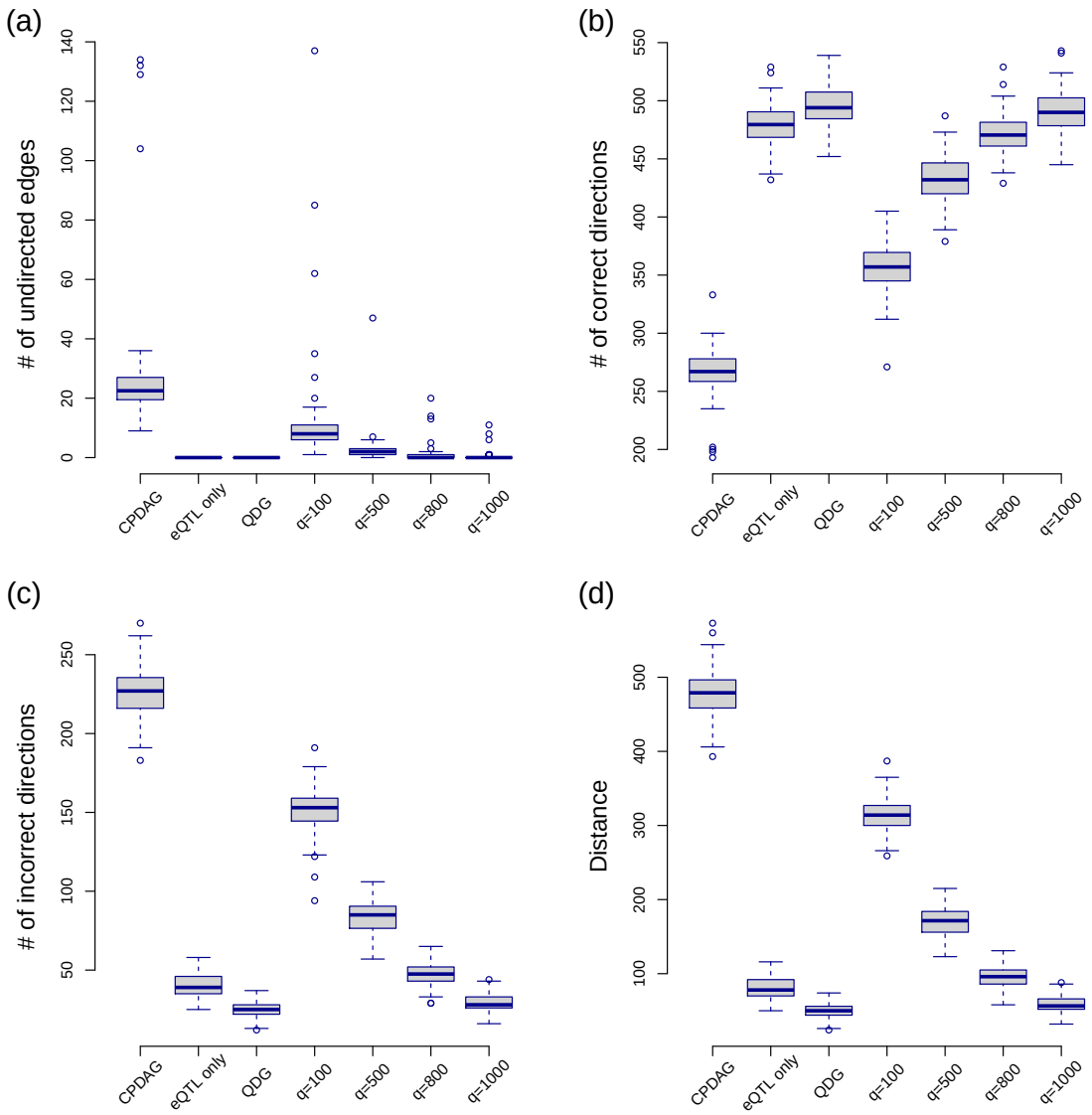

Figure 5.3: Performances of CPDAG, directions when only eQTLs are used to calculate likelihoods, QDG, `siDAG` for q=100, 500, 800 and 1000 when p=1000, n=300, $p_m = 0.3$ and $p_E = 0.002$. Among true positive undirected edges in the skeleton estimates (a) number of undirected edges. (b) number of correct direction (c) number of incorrect directions. (d) `Distance`.

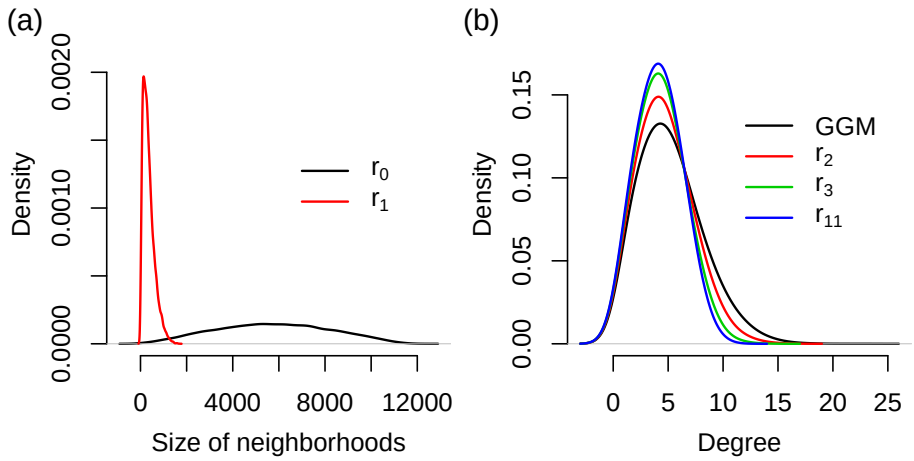

Figure 5.4: (a) Distribution of neighborhood sizes after the pre-screening procedure. (b) Distribution of degree during PC-algorithm

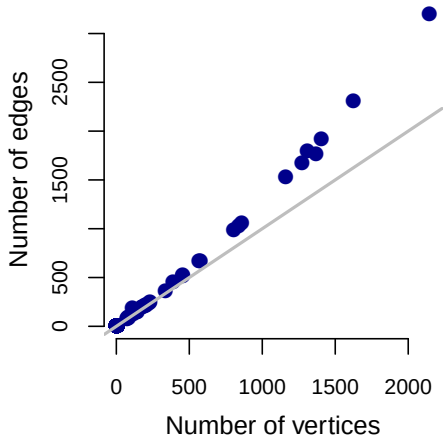

Figure 5.5: A scatter plot of the number of vertices versus the number of edges in each module

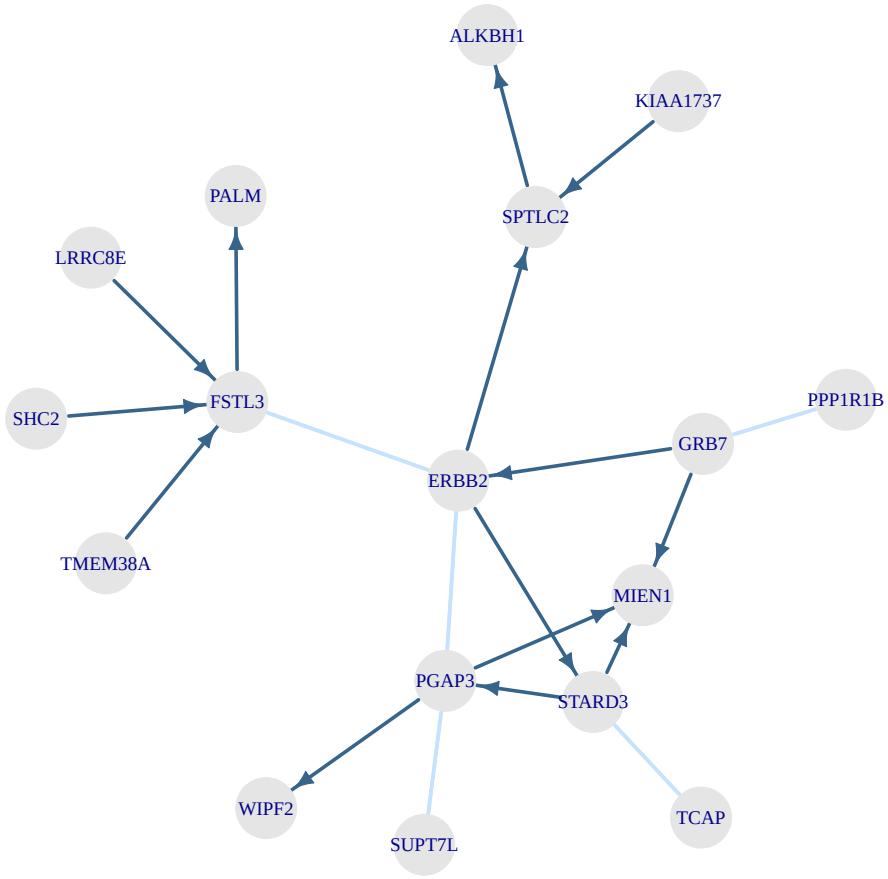

Figure 5.6: DAG estimation around gene ERBB2, where light blue edges are un-directed edges.

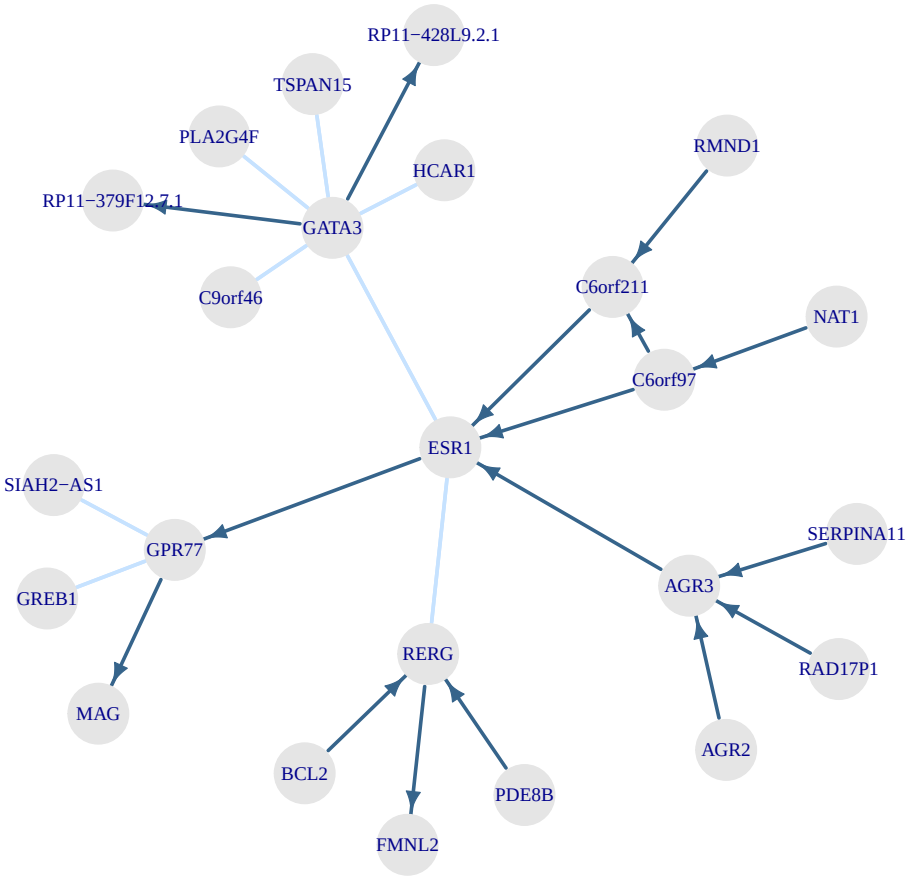

Figure 5.7: DAG estimation around gene ESR1, where light blue edges are un-directed edges.

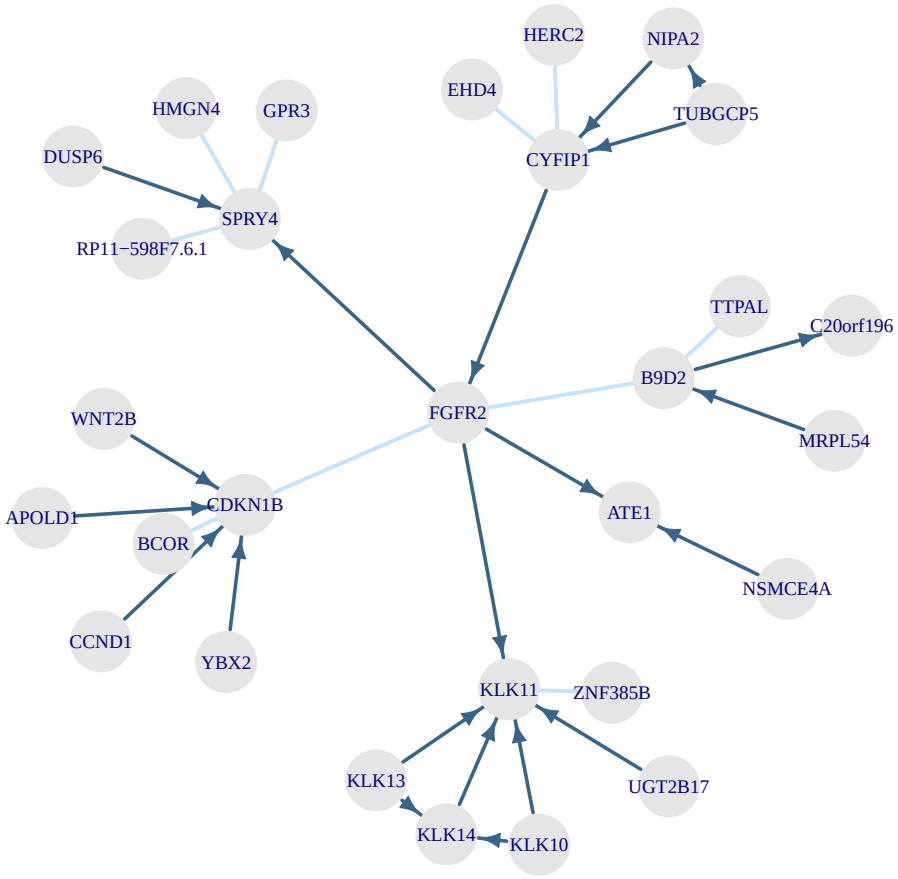

Figure 5.8: DAG estimation around gene FGFR2, where light blue edges are un-directed edges.

Appendix I

Supplementary materials for Chapter 3

Inverse covariance matrix and partial correlation matrix

Consider a p-dimensional random variable $X = (X_1, ..., X_p)^{\mathrm{T}}$, with mean zero and $p \times p$ positive definite covariance matrix $\mathbf{\Sigma}$. Writing $Y = (X_a, X_b)^{\mathrm{T}}$ and $Z = X_{\mathbf{\Gamma}\backslash\{a,b\}} = \{X_k : k \in \mathbf{\Gamma}\backslash\{a,b\}\}$. Suppose the random vector $(Y, Z)^{\mathrm{T}}$ have covariance matrix $\tilde{\mathbf{\Sigma}}$, which is a permutation of the covariance matrix $\mathbf{\Sigma}$. Let $\tilde{\mathbf{\Omega}} = \tilde{\mathbf{\Sigma}}^{-1}$. $\tilde{\mathbf{\Sigma}}$ and $\tilde{\mathbf{\Omega}}$ can be partitioned as

$$\tilde{\mathbf{\Sigma}} = \begin{pmatrix} \mathbf{\Sigma}_{YY} & \mathbf{\Sigma}_{YZ} \\ \mathbf{\Sigma}_{ZY} & \mathbf{\Sigma}_{ZZ} \end{pmatrix}, \quad \tilde{\mathbf{\Omega}} = \begin{pmatrix} \mathbf{\Omega}_{YY} & \mathbf{\Omega}_{YZ} \\ \mathbf{\Omega}_{ZY} & \mathbf{\Omega}_{ZZ} \end{pmatrix}.$$

Given the formula for the inverse of a partitioned matrix,

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} E^{-1} & -E^{-1}G \\ -FE^{-1} & D^{-1} + FE^{-1}G \end{pmatrix},$$

where $E = A - BD^{-1}C$, $F = D^{-1}C$, and $G = BD^{-1}$, it is easy to show that

$$\mathbf{\Omega}_{YY} = (\mathbf{\Sigma}_{YY} - \mathbf{\Sigma}_{YZ}\mathbf{\Sigma}_{ZZ}^{-1}\mathbf{\Sigma}_{ZY})^{-1}. \tag{5.11}$$

Now we show that $-\texttt{scale}(\mathbf{\Omega}_{YY})$ is the partial correlation between X_a and X_b. The best linear predictor of Y given Z is $\hat{\boldsymbol{\beta}}^{\mathrm{T}}$ where $\hat{\boldsymbol{\beta}}$ is the solution of $\mathbf{\Sigma}_{ZZ}\hat{\boldsymbol{\beta}} = \mathbf{\Sigma}_{ZY}$. The partial covariance matrix of Y, which is defined as the covariance of the residuals, is

$$\texttt{Cov}(Y - \hat{\boldsymbol{\beta}}^{\mathrm{T}}Z) = \mathbf{\Sigma}_{YY} - \hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{\Sigma}_{ZZ}\hat{\boldsymbol{\beta}} = \mathbf{\Sigma}_{YY} - \mathbf{\Sigma}_{YZ}\mathbf{\Sigma}_{ZZ}^{-}\mathbf{\Sigma}_{ZY} \tag{5.12}$$

for any generalized inverse, $\mathbf{\Sigma}_{ZZ}^{-}$ that satisfies $\mathbf{\Sigma}_{ZZ} = \mathbf{\Sigma}_{ZZ}\mathbf{\Sigma}_{ZZ}^{-}\mathbf{\Sigma}_{ZZ}$.

Comparing equations (5.11) and (5.12), we have

$$\texttt{Cov}(y - \hat{\boldsymbol{\beta}} Z) = \boldsymbol{\Omega}_{YY}^{-1} = \frac{1}{\texttt{det}(\boldsymbol{\Omega}_{YY})} \begin{pmatrix} \boldsymbol{\Omega}_{bb} & -\boldsymbol{\Omega}_{ab} \\ -\boldsymbol{\Omega}_{ba} & \boldsymbol{\Omega}_{aa} \end{pmatrix}.$$

By normalizing the matrix, we get the partial correlation of X_a and X_b as

$$\frac{-\boldsymbol{\Omega}_{ab}}{\sqrt{\boldsymbol{\Omega}_{aa}}\sqrt{\boldsymbol{\Omega}_{bb}}}.$$

which is off diagonal element of $-\texttt{scale}(\boldsymbol{\Omega}_{YY})$. This result can be generalized to any pair of nodes.

Appendix II

Supplementary materials for Chapter 4

An example that neither covariance matrix nor concentration matrix captures the network skeleton

Consider a simple network of four nodes/variables X, Y, and Z and W, with the underlying network structure $X \to W \leftarrow Z \leftarrow Y$, and we assume there is no any other (hidden) variables. For illustration purpose, we assume the observations of these four random variables are generated through the following mechanism.

$$X = \epsilon_1,\ Y = \epsilon_2,\ Z \ = \ Y + \epsilon_3,\ \text{and } W = X + Z + \epsilon_4 \tag{5.13}$$

where ϵ_j are i.i.d. $N(0,1)$ for $1 \le j \le 4$. Denote the covariance matrix and partial covariance matrix of this system as $\mathbf{\Sigma}$ and $\mathbf{\Omega}$, respectively. Note $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$, and (i,j)-th entry of $\mathbf{\Omega}$ indicates the covariance of the i-th and the j-th variables, conditioning on all the other covariates in this system. Let the connection matrix (i.e., skeleton) of this system be $\mathbf{\Xi}$. Then we have:

$$\mathbf{\Sigma} = \begin{array}{c} \\ X \\ Y \\ Z \\ W \end{array} \begin{array}{c} \begin{array}{cccc} X & Y & Z & W \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 \\ 1 & 1 & 2 & 4 \end{pmatrix} \end{array}, \ \mathbf{\Omega} = \begin{array}{c} \\ X \\ Y \\ Z \\ W \end{array} \begin{array}{c} \begin{array}{cccc} X & Y & Z & W \end{array} \\ \begin{pmatrix} 2 & 0 & 1 & -1 \\ 0 & 2 & -1 & 0 \\ 1 & -1 & 2 & -1 \\ -1 & 0 & -1 & 1 \end{pmatrix} \end{array}, \text{ and } \mathbf{\Xi} = \begin{array}{c} \\ X \\ Y \\ Z \\ W \end{array} \begin{array}{c} \begin{array}{cccc} X & Y & Z & W \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array}.$$

We see that neither $\mathbf{\Sigma}$ nor $\mathbf{\Omega}$ gives us the correct connection matrix of network structure $X \to W \leftarrow Z \leftarrow Y$.

The details of the `PenPC` algorithm

In this section, we describe the step 2 of `PenPC` algorithm. For any undirected graph $\mathbf{C} = (V, F)$ we define some quantities. Denote $\texttt{adj}(i, \mathbf{C})$ as the adjacent vertices

of i in $\mathbf{C}$. Let $\mathtt{adj}(i,j,\mathbf{C}) = \mathtt{adj}(i,\mathbf{C}) \bigcap \mathtt{adj}(j,\mathbf{C})$. The subgraph of $\mathbf{C}$ on the set of vertices $S \subseteq V$ is denoted by $\mathbf{C}(S)$. And $\mathtt{Con}(v, \mathbf{C}(S))$ for $v \in S$ is defined by the set of connected vertices to v by any length of chains in $\mathbf{C}(S)$ and includes v itself. For an edge $(i,j) \in F$, $\mathbf{\Gamma}(\mathbf{C})_{i,j}$ is defined by

$$\mathbf{\Gamma}(\mathbf{C})_{i,j} = \left[\bigcup_{v \in \mathtt{adj}(i,j,\mathbf{C})} \mathtt{Con}\big(v, \mathbf{C}(V \setminus \{i,j\})\big) \right] \bigcap \left[\mathtt{adj}(i,\mathbf{C}) \bigcup \mathtt{adj}(j,\mathbf{C}) \right].$$

Let $\mathcal{C}_{\mathcal{G}} = (V, F_{\mathcal{G}})$ as the GGM in step 1 of PenPC algorithm described in section 4.3.

Input: GGM $\mathcal{C}_{\mathcal{G}}$

Output: Skeleton $\mathcal{G}^u = (V, E^u)$ and separation set $S(i,j)$ for edges $(i,j) \notin E^u$ but $(i,j) \in F_{\mathcal{G}}$

1. **Set** l=-1 and $\mathbf{C} = \mathcal{C}_{\mathcal{G}}$ $(F = F_{\mathcal{G}})$
2. **For** all $(i,j) \in F$,
   - 2.1 if X_i and X_j are marginally independent, then delete (i,j) from F
3. **Repeat:** l=l+1
   - 3.1 **Repeat:** Select an edge $(i,j) \in F$ such that $|\mathbf{\Gamma}(\mathbf{C})_{i,j}| \geq l$
     - 3.1.1 **Repeat:** Select $\Gamma \subseteq \mathbf{\Gamma}(\mathbf{C})_{i,j}$ with $|\Gamma| = l$
       - 3.1.1.1 Set $\mathcal{K} = [\mathtt{adj}(i,\mathbf{C}) \cup \mathtt{adj}(j,\mathbf{C})] \backslash [\Gamma \bigcup \{i,j\}]$
       - 3.1.1.2 If X_i and X_j are conditionally independent given $\{X_k : k \in \mathcal{K}\}$, then
         - Delete (i,j) from F
         - Save $\mathcal{K}$ in separation set for i and j, $S(i,j)$

3.1.2 **Until:** The edge (i, j) is deleted from F or all $|\Gamma| = l$ have been chosen

3.2 **Until:** All edges $(i, j) \in F$ with $|\mathbf{\Gamma}(\mathbf{C})_{i,j}| \geq l$ are tested for all conditioning set $\Gamma \subseteq \mathbf{\Gamma}(\mathbf{C})_{i,j}$ with $|\Gamma| = l$

4. **Until:** for each $(i, j) \in F$, $|\mathbf{\Gamma}(\mathbf{C})_{i,j}| < l$

The deterministic rules to extend a skeleton to a CPDAG

We describe the rules in [Kalisch and Bühlmann, 2007] and [Pearl, 2009]. Given the skeleton $\mathcal{G}^u$ and the separation sets $S(i, j)$ for all missing edges between nodes i and j, the arrow orientation of the skeleton proceeds in two step: (1) determination of the v-structure and (2) completion of the partially directed graph (PDAG) in (1).

step 1 For each pair of nonadjacent vertices i and j with common neighbor k, add arrow heads pointing at k, $i \to k \leftarrow j$ if $k \notin S(i, j)$.

step 2 In the PDAG from step 1, following four rules are repeatedly applied to obtain maximally oriented pattern.

rule 1 Orient $j - k$ into $j \to k$ whenever there is an arrow $i \to j$ such that i and k are nonadjacent.

rule 2 Orient $i - j$ into $i \to j$ whenever there is a path $i \to k \to j$.

rule 3 Orient $i - j$ into $i \to j$ whenever there are two paths $i - k_1 \to j$ and $i - k_2 \to j$ such that k_1 and k_2 are nonadjacent.

rule 4 Orient $i - j$ into $i \to j$ whenever there are two paths $i - k_1 \to k_2$ and $k_1 \to k_2 \to j$ such that k_1 and k_2 are nonadjacent.

Supplementary Theoretical Results

Proof of Lemma 1

Proof. Without loss of generality, let $i = 1$. Partition the covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix},$$

where Σ_{11} is a scaler and $\boldsymbol{\Sigma}_{22}$ is a $(p-1) \times (p-1)$ matrix.

It is easy to show that

$$X_1 \mid \boldsymbol{X}_{-1} \sim \mathrm{N}\left(\boldsymbol{X}_{-1}^{\mathrm{T}}(\boldsymbol{\Sigma}_{22})^{-1}\boldsymbol{\Sigma}_{21}, \Sigma_{11} - \boldsymbol{\Sigma}_{12}(\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})\right).$$

Therefore, $\boldsymbol{b}_i = (\boldsymbol{\Sigma}_{22})^{-1}\boldsymbol{\Sigma}_{21}$

On the other hand, by block matrix inversion formula,

$$\boldsymbol{\Omega} = \begin{pmatrix} \Sigma_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{\sigma_1^2} & -\frac{1}{\sigma_1^2}\boldsymbol{b}_i^{\mathrm{T}} \\ -\frac{1}{\sigma_1^2}\boldsymbol{b}_i & (\boldsymbol{\Sigma}_{22} - \frac{1}{\Sigma_{11}}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{21})^{-1} \end{pmatrix},$$

where $\sigma_1^2 = \Sigma_{11} - \boldsymbol{\Sigma}_{12}(\boldsymbol{\Sigma}_{22})^{-1}\boldsymbol{\Sigma}_{21}$. □

Lemma 5

The following lemma is needed for proof of Theorem 2. It provides a sufficient condition for strict local minimizer $\hat{\boldsymbol{b}}_i$ of (4.1).

Lemma 5. *Assume that $p_{\boldsymbol{\theta}}$ satisfies Condition 1. Define $\bar{p}(t) = sgn(t)p'_{\boldsymbol{\theta}}(|t|)$, $t \in \mathbb{R}$ and $\bar{p}(\boldsymbol{t}) = (\bar{p}(t_1), \ldots, \bar{p}(t_q))$, $\boldsymbol{t} = (t_1, \ldots, t_q)^{\mathrm{T}}$. Then $\hat{\boldsymbol{b}}_i \in \mathbb{R}^{p_n - 1}$ is a strict local minimizer*

of $\boldsymbol{Q}(\boldsymbol{b}_i) = \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{X}_{-i}\boldsymbol{b}_i)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{X}_{-i}\boldsymbol{b}_i) + n\sum_{j\neq i} p_{\boldsymbol{\theta}}(|b_{i,j}|)$ *if*

$$\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{X}_{-i}\hat{\boldsymbol{b}}_i) = n\bar{p}(\hat{\boldsymbol{b}}_{i1}), \tag{5.14}$$

$$\|\boldsymbol{\mathcal{X}}_{i2}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{X}_{-i}\hat{\boldsymbol{b}}_i)\|_\infty < np'_{\boldsymbol{\theta}}(0+), \tag{5.15}$$

$$\|(\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1})^{-1}\|_2 \leq 1/(n\kappa(p;\hat{\boldsymbol{b}}_{i1})) \tag{5.16}$$

where $\kappa(p;\hat{\boldsymbol{b}}_{i1})$ *is defined in (4.2).*

It is a special case of Theorem 1 in [Fan and Lv, 2011], which provides more detail discussion about the conditions so that we skip the proof.

Proof of Theorem 2

Proof. For any fixed $i \in V_n$, $\boldsymbol{x}_i$ is a $n \times 1$ response vector and $\boldsymbol{X}_{-i}$ is a $n \times q$ covariate matrix with $q = p_n - 1$ corresponding to vertices $V_n \setminus \{i\}$. Let $\mathcal{S}_i = \mathrm{supp}(\boldsymbol{b}_i)$ to be the support of the true regression coefficient $\boldsymbol{b}_i$ with $|\mathcal{S}_i| = s_i$. Define $\boldsymbol{\xi}_i = (\xi_{i1}, ..., \xi_{iq})^{\mathrm{T}} = \boldsymbol{X}_{-i}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{X}_{-i}\boldsymbol{b}_i) = \boldsymbol{X}_{-i}^{\mathrm{T}}\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim N_n(0, \sigma_i^2 I_n)$ for $n \times n$ identity matrix I_n. Let $\boldsymbol{\xi}_{i1}$ and $\boldsymbol{\xi}_{i2}$ to be the non-joint sub-vectors with indices partitioned by $\mathcal{S}_i$. Define the event

$$\mathcal{E}_i = \left\{ \|\boldsymbol{\xi}_i\|_\infty \leq \sigma_i n^{1/2+a/2}\sqrt{\log(n)} \right\}. \tag{5.17}$$

We first consider the property of penalized regression on $\mathcal{E}_i$. Lemma 5 gives out sufficient conditions of a local minimizer. We prove that within the hypercube

$$\mathcal{N}_i = \{\boldsymbol{\beta} = (\boldsymbol{\beta}_1^{\mathrm{T}}, \boldsymbol{\beta}_2^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^q : \|\boldsymbol{\beta}_1 - \boldsymbol{b}_{i1}\|_\infty \leq Cn^{-d_2},\ \boldsymbol{\beta}_2 = 0\}, \tag{5.18}$$

there is a solution $\hat{\boldsymbol{b}}_i$ that satisfy (5.14) (5.15) and (5.16). Equation (5.16) of Lemma 5 holds by Assumption (A6).

Step 1: Find a solution to (5.14) in $\mathcal{N}_i$.

We will prove that conditioning on $\mathcal{E}_i$, there is a solution $\hat{\boldsymbol{b}}_{i1} \in \mathbb{R}^{s_i}$ for equation (5.14) of Lemma 5 which is equivalent to

$$\hat{\boldsymbol{b}}_{i1} = \boldsymbol{b}_{i1} + (\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1})^{-1}\{\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}\boldsymbol{\epsilon} - n\bar{p}(\hat{\boldsymbol{b}}_{i1})\}.$$

Suppose that $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^{\mathrm{T}}, \boldsymbol{\beta}_2^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^q$ has the same partition as $\boldsymbol{b}_i = (\boldsymbol{b}_{i1}^{\mathrm{T}}, \boldsymbol{b}_{i2}^{\mathrm{T}})^{\mathrm{T}}$. Let $\boldsymbol{u}_i = (\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1})^{-1}[\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}\boldsymbol{\epsilon} - n\bar{p}(\boldsymbol{\beta}_1)]$, and $\phi(\boldsymbol{\beta}_1) = \boldsymbol{\beta}_1 - \boldsymbol{b}_{i1} - \boldsymbol{u}_i$, where $\boldsymbol{\beta}_1 = (\beta_{1,1}, \ldots, \beta_{1,s_i})^{\mathrm{T}} \in \mathbb{R}^{s_i}$ and $\boldsymbol{b}_{i1} = (b_{i1,1}, \ldots, b_{i1,s_i}) \in \mathbb{R}^{s_i}$. It suffies to show that there is a solution to $\phi(\boldsymbol{\beta}_1) = \boldsymbol{0}$ in $\mathcal{N}_i$. Suppose $\|\boldsymbol{u}_i\|_\infty = o(n^{-d_2})$. For sufficiently large n, if $\beta_{1,j} - b_{i1,j} = Cn^{-d_2}$, $\phi_j(\boldsymbol{\beta}_1) \geq Cn^{-d_2} - \|\boldsymbol{u}_i\|_\infty > 0$. If $\beta_{1,j} - b_{i1,j} = -Cn^{-d_2}$, $\phi_j(\boldsymbol{\beta}_1) \leq -Cn^{-d_2} + \|\boldsymbol{u}_i\|_\infty < 0$. By the continuity of function $\phi(\boldsymbol{\beta}_1)$ and Miranda's existence theorem, there is a solution for $\phi(\boldsymbol{\beta}_1) = \boldsymbol{0}$ in $\mathcal{N}_i$.

Now we prove $\|\boldsymbol{u}_i\|_\infty = o(n^{-d_2})$. For any $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^{\mathrm{T}}, \boldsymbol{\beta}_2^{\mathrm{T}})^{\mathrm{T}} \in \mathcal{N}_i$, $|\beta_{1,j}| \geq |b_{i1,j}| - \delta_n$ where δ_n defined in Assumption (A5), and thus

$$\min_{j=1,\ldots,s_i} |\beta_{1,j}| \geq \min_{j=1,\ldots,s_i} |b_{i1,j}| - \delta_n \geq \delta_n.$$

By monotonicity of $p'_{\boldsymbol{\theta}}(t)$ in Condition 1, $\|\bar{p}_{\boldsymbol{\theta}}(\boldsymbol{\beta}_1)\|_\infty \leq p'_{\boldsymbol{\theta}}(\delta_n)$. Therefore, on $\mathcal{E}_i$,

$$\|\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}\boldsymbol{\epsilon} - np'_{\boldsymbol{\theta}}(\boldsymbol{\beta}_1)\|_\infty \leq \sigma_i n^{1/2+a/2}\sqrt{\log(n)} + np'_{\boldsymbol{\theta}}(\delta_n),$$

Then by assumption (A4),

$$\|\boldsymbol{u}_i\|_\infty \leq \sigma_i n^{-1/2+a/2+s_0}\sqrt{\log(n)} + n^{s_0}p'_{\boldsymbol{\theta}}(\delta_n)$$

By Assumption (A5), $\sigma_i n^{-1/2+a/2+s_0}\sqrt{\log(n)} = o(n^{-d_2})$ and by Assumption (A6), $n^{s_0}p'_{\boldsymbol{\theta}}(\delta_n) = o(n^{-d_2})$. Therefore, $\|\boldsymbol{u}_i\|_\infty = o(n^{-d_2})$. From this proof, we require the penalty to be small enough $p'_{\boldsymbol{\theta}}(\delta_n) = o(n^{-s_0-d_2})$.

Step 2: Verify Condition (5.15) holds for $\hat{\boldsymbol{b}}_i$.

For $\hat{\boldsymbol{b}}_i \in \mathcal{N}_i$ satisfying the condition (5.14), we need to verify

$$\|\boldsymbol{\mathcal{X}}_{i2}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{X}_{-i}\hat{\boldsymbol{b}}_i)\|_\infty < np'_{\boldsymbol{\theta}}(0+)$$

on the event $\mathcal{E}_i$. Note that

$$\boldsymbol{\mathcal{X}}_{i2}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{X}_{-i}\hat{\boldsymbol{b}}_i) = \boldsymbol{\mathcal{X}}_{i2}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{X}_{-i}\boldsymbol{b}_i) - \boldsymbol{\mathcal{X}}_{i2}^{\mathrm{T}}(\boldsymbol{X}_{-i}\hat{\boldsymbol{b}}_i - \boldsymbol{X}_{-i}\boldsymbol{b}_i) = \boldsymbol{\xi}_{i2} - \boldsymbol{\mathcal{X}}_{i2}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1}(\hat{\boldsymbol{b}}_{i1} - \boldsymbol{b}_{i1}).$$

By Condition 1, $n\|p'_{\boldsymbol{\theta}}(\hat{\boldsymbol{b}}_{i1})\|_\infty \leq np'_{\boldsymbol{\theta}}(\delta_n)$. From assumption (A5) and (A6) we know that $n\|\boldsymbol{\mathcal{X}}_{i2}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1}(\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1})\|_\infty p'_{\boldsymbol{\theta}}(\delta_n) \leq Knp'_{\boldsymbol{\theta}}(0+)$ and $n^{1/2+a/2+b}\sqrt{\log(n)} = o(np'_{\boldsymbol{\theta}}(0+))$. On $\mathcal{E}_i$,

$$\begin{aligned}
&\|\boldsymbol{\mathcal{X}}_{i2}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{X}_{-i}\hat{\boldsymbol{b}}_i)\|_\infty \\
\leq&\|\boldsymbol{\xi}_{i2}\|_\infty + \|\boldsymbol{\mathcal{X}}_{i2}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1}(\hat{\boldsymbol{b}}_{i1} - \boldsymbol{b}_{i1})\|_\infty \\
\leq&\sigma_i n^{1/2+a/2}\sqrt{\log(n)} + \|\boldsymbol{\mathcal{X}}_{i2}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1}(\boldsymbol{\mathcal{X}}_{i1}^{\mathrm{T}}\boldsymbol{\mathcal{X}}_{i1})^{-1}\|_\infty \left[\|\boldsymbol{\xi}_{i1}\|_\infty + n\|p'_{\boldsymbol{\theta}}(\hat{\boldsymbol{b}}_{i1})\|_\infty\right] \\
\leq&o(np'_{\boldsymbol{\theta}}(0+)) + \sigma_i n^{1/2+a/2+b}\sqrt{\log(n)} + Knp'_{\boldsymbol{\theta}}(0+) \\
<&np'_{\boldsymbol{\theta}}(0+)
\end{aligned}$$

for sufficiently large n.

Step 3: Prove that $\mathbb{P}(\mathcal{E}_i) \to 1$.

Since $\|\boldsymbol{x}_i\|_2 = \sqrt{n}$, $(\sqrt{n}\sigma_i)^{-1}\xi_{ij} \sim \mathrm{N}(0,1)$. Applying the upper bound for normal distribution function,

$$\mathbb{P}\left(\frac{\xi_{ij}}{\sqrt{n}\sigma_i} > z\right) < \frac{1}{\sqrt{2\pi}}z^{-1}\exp\{-z^2/2\}.$$

We have

$$\begin{aligned}
\mathbb{P}(\mathcal{E}_i) &\geq 1 - \sum_{j=1}^{q} \mathbb{P}\left(|\xi_{ij}| > \sigma_i n^{1/2+a/2}\sqrt{\log(n)}\right) \\
&> 1 - \sum_{j=1}^{q} \frac{\sqrt{2}}{\sqrt{\pi n^a \log(n)}} \exp\{-n^a \log(n)/2\} \\
&> 1 - Cp_n \exp\{-n^a \log(n)/2\} \\
&> 1 - C\exp\{n^a - n^a \log(n)/2.
\end{aligned}$$

□

Proof of Corollary 1

Proof. Let $\mathcal{E} = \bigcap_{i=1}^{p_n} \mathcal{E}_i$ where $\mathcal{E}_i$ defined in (5.17). Therefore $\mathbb{P}(\mathcal{E}) \geq 1 - \sum_{i=1}^{p_n}(1 - \mathbb{P}(\mathcal{E}_i)) \geq 1 - C\exp\{2n^a - n^a \log(n)/2\} \to 1$ from the proof of Theorem 2. □

Proof of Lemma 2

Proof. We need to show that $(i,j) \in F_n \Leftrightarrow (i,j) \in E_n^u$ or at least one vertex k such that $i \to k \leftarrow j$. If $(i,j) \in E_n^u$, i and j are conditionally dependent given all subsets of $V_n \setminus \{i,j\}$, hence $(i,j) \in F_n$. If i and j are common parents of a common child k, i and j are conditionally dependent given $V_n \setminus \{i,j\}$ since $k \in V_n \setminus \{i,j\}$. Conversely, if $(i,j) \in F_n$, i and j are not separated by $V_n \setminus \{i,j\}$. By Definition 1 on d-separation, it means $(i,j) \in E_n^u$ or some of the chains between i and j are not d-separated by $V_n \setminus \{i,j\}$. The latter case means that at least one chain has a collider (or a v-structure). It is easy to show if this chain includes any vertices other than the collider, it is d-separated by $V_n \setminus \{i,j\}$. Therefore i and j co-parent at least one child. □

Proof of Lemma 3

Proof. Suppose that two vertices i and j are not connected in the skeleton $\mathcal{G}_n^u$, but connected in the GGM $\mathcal{C}_{\mathcal{G}_n}$. By Lemma 2, there exists at least one vertex k such that $i \to k \leftarrow j$. Let $\texttt{adj}_j(i, \mathcal{C}_{\mathcal{G}_n})$ as adjacency vertices of i without j in $\mathcal{C}_{\mathcal{G}_n}$. Let $\texttt{ch}_{\mathcal{G}_n}(i)$ and $\texttt{de}_{\mathcal{G}_n}(i)$ be the sets of children and descendants of i in $\mathcal{G}_n$. Let $\pi_i = \texttt{adj}_j(i, \mathcal{C}_{\mathcal{G}_n}) \setminus \left[\bigcup_{v \in \texttt{ch}_{\mathcal{G}_n}(i) \cap \texttt{ch}_{\mathcal{G}_n}(j)} (\{v\} \bigcup \texttt{de}_{\mathcal{G}_n}(v))\right]$ and $\pi_j = \texttt{adj}_i(j, \mathcal{C}_{\mathcal{G}_n}) \setminus \left[\bigcup_{v \in \texttt{ch}_{\mathcal{G}_n}(i) \cap \texttt{ch}_{\mathcal{G}_n}(j)} (\{v\} \bigcup \texttt{de}_{\mathcal{G}_n}(v))\right]$. We show that i and j are d-separated by $\pi_i \bigcup \pi_j$ and $\pi_i \bigcup \pi_j \in \Pi_{i,j}$.

In order to show that $i \perp\!\!\!\perp j | \pi_i \bigcup \pi_j$, we consider a sequence of vertices $k_1, \ldots, k_m$ for $m \geq 1$ of chains such that

(Chain 1) $i \to k_1 - \ldots - k_m \leftarrow j$

(Chain 2) $i \to k_1 - \ldots - k_m \to j$

(Chain 3) $i \leftarrow k_1 - \ldots - k_m \leftarrow j$

(Chain 4) $i \leftarrow k_1 - \ldots - k_m \to j$.

Those four cases cover all possible chains connecting i and j for the adjacent vertices, k_1 and k_m. It suffices to show that $\pi_i \bigcup \pi_j$ blocks all the four types of chains between i and j. For the (Chain 2), a set including the arrow emitting vertex k_m d-separates i and j by Definition 1 on d-separation. Since $k_m \in \texttt{adj}_i(j, \mathcal{C}_{\mathcal{G}_n})$ and $k_m \notin \texttt{ch}_{\mathcal{G}_n}(j)$ because of no loop, $k_m \in \pi_i \bigcup \pi_j$. Similarly for the (Chain 3), since the arrow emitting vertex $k_1 \in \texttt{adj}_j(i, \mathcal{C}_{\mathcal{G}_n})$ but $k_1 \notin \texttt{ch}_{\mathcal{G}_n}(i)$, $k_1 \in \pi_i \bigcup \pi_j$. The (Chain 4) also blocked by either arrow-emitting vertices k_1 or k_2 included in $\pi_i \bigcup \pi_j$. In the (Chain 1), there must be at least one collider. If $m = 1$, then k_m is a common child so that it is excluded from $\pi_i \bigcup \pi_j$. If $m = 2$, the possible chains are $i \to k_1 \to k_2 \leftarrow j$ or $i \to k_1 \leftarrow k_2 \leftarrow j$ and both chains have one arrow emitting vertex, k_1 or k_2 in $\pi_i \bigcup \pi_j$. Now we suppose that there are at least three vertices, $m > 2$. If at least one of k_1 and k_m is not a collider, there exists a arrow emitting vertex in $\pi_i \bigcup \pi_j$. If both k_1 and k_m are colliders, the (Chain 1) is $i \to k_1 \leftarrow k_2 - \ldots - k_{m-1} \to k_m \leftarrow j$. Since the arrow emitting vertices

k_2 and k_{m-1} are not in $\mathtt{ch}_{\mathcal{G}_n}(i) \bigcap \mathtt{ch}_{\mathcal{G}_n}(j)$ but in $\mathtt{adj}_j(i, \mathcal{C}_{\mathcal{G}_n}) \bigcup \mathtt{adj}_i(j, \mathcal{C}_{\mathcal{G}_n})$, those are in $\pi_i \bigcup \pi_j$. Therefore, $\pi_i \bigcup \pi_j$ blocks all chains between i and j.

Next we need to prove $\pi_i \bigcup \pi_j \in \Pi_{i,j}$. Let $V_{n,-i,-j} = V_n \setminus \{i, j\}$. Since $\pi_i \bigcup \pi_j = \left[\mathtt{adj}_j(i, \mathcal{C}_{\mathcal{G}_n}) \bigcup \mathtt{adj}_i(j, \mathcal{C}_{\mathcal{G}_n})\right] \setminus \left[\bigcup_{v \in \mathtt{ch}_{\mathcal{G}_n}(i) \cap \mathtt{ch}_{\mathcal{G}_n}(j)} \left(\{v\} \bigcup \mathtt{de}_{\mathcal{G}_n}(v)\right)\right]$, it suffices to show that

$$\bigcup_{v \in \mathtt{ch}_{\mathcal{G}_n}(i) \cap \mathtt{ch}_{\mathcal{G}_n}(j)} \Big(\{v\} \bigcup \mathtt{de}_{\mathcal{G}_n}(v)\Big) \subseteq \bigcup_{v \in \mathtt{adj}(i,j,\mathcal{C}_{\mathcal{G}_n})} \mathtt{Con}\big(v, \mathcal{C}_{\mathcal{G}_n}(V_{n,-i,-j})\big). \tag{5.19}$$

First, $\mathtt{ch}_{\mathcal{G}_n}(i) \bigcap \mathtt{ch}_{\mathcal{G}_n}(j) \subseteq \mathtt{adj}(i, j, \mathcal{C}_{\mathcal{G}_n})$ because $\mathtt{adj}(i, j, \mathcal{C}_{\mathcal{G}_n})$ contains all common children as well as all common parents of i and j. Second, for a fixed vertex $v \in \mathtt{adj}(i, j, \mathcal{C}_{\mathcal{G}_n})$, the set of vertices connected to v by any length of chains in the subgraph $\mathcal{C}_{\mathcal{G}_n}(V_{n,-i,-j})$ includes all the descendents of v if v is a common child of i and j. Therefore the relationship (5.19) holds.

If we test i and j conditioning on each set in $\Pi_{i,j}$ made by excluding all subsets of $\bigcup_{v \in \mathtt{adj}(i,j,\mathcal{C}_{\mathcal{G}_n})} \mathtt{Con}\big(v, \mathcal{C}_{\mathcal{G}_n}(V_{n,-i,-j})\big)$ from the union of the two adjacent vertices, $\mathtt{adj}_j(i, \mathcal{C}_{\mathcal{G}_n}) \bigcup \mathtt{adj}_i(j, \mathcal{C}_{\mathcal{G}_n})$, we can always eliminate the edge $(i, j) \in F_n$ of $\mathcal{C}_{\mathcal{G}_n}$ being in co-parent relationship in $\mathcal{G}_n$. □

Lemma 6

We state Lemma 6 which is used to prove Theorem 3. This lemma is essentially the same as Lemma 3 in [Kalisch and Bühlmann, 2007]. The proof is therefore skipped. Let $\nu_i = |\mathtt{adj}(i, \mathcal{C}_{\mathcal{G}_n})|$ for all $i \in V_n$.

Lemma 6. *Let* $g(\rho) = 0.5 \log((1 + \rho)/(1 - \rho))$. *Denote by* $\hat{z}_{i,j|\mathcal{K}} = g\left(\hat{\rho}_{i,j|\mathcal{K}}\right)$ *and by* $z_{i,j|\mathcal{K}} = g\left(\rho_{i,j|\mathcal{K}}\right)$ *where* $\mathcal{K} \subseteq \mathtt{adj}(i, \mathcal{C}_{\mathcal{G}_n}) \bigcup \mathtt{adj}(j, \mathcal{C}_{\mathcal{G}_n})$. *Assume the distribution of* $X = (X_1, X_2, ..., X_p)^{\mathrm{T}}$ *is multivariate Gaussian (the frist part of Assumption (A1)), and* $\sup_{i,j,\mathcal{K}} \left|\rho_{i,j|\mathcal{K}}\right| \leq M < 1$ *(the second part of Assumption (A3)). Then, for any*

$0 < \gamma < 2$,

$$\sup_{i,j,\mathcal{K}} \mathbf{P}\left(\left|\hat{z}_{i,j|\mathcal{K}} - z_{i,j|\mathcal{K}}\right| > \gamma\right) \leq O(n - \nu_i - \nu_j)\left[\exp\left\{-(C_1 + C_2)(n - \nu_i - \nu_j - 4)\right\}\right],$$

where $0 < C_1 < \infty$, *and* $0 < C_2 < \infty$. *More specifically,*

$$C_1 = \log\left[\frac{4 + (\gamma l)^2}{4 - (\gamma l)^2}\right], \quad C_2 = \log\left[\frac{16 + (1 - M)^2}{16 - (1 - M)^2}\right],$$

where $l = 1 - (1 + M)^2/4$.

Proof of Theorem 3

Proof of Theorem 3. For an edge $(i, j) \in F_n$ of $\mathcal{C}_{\mathcal{G}_n}$, define $\mathcal{K}$ to be any set in $\Pi_{i,j}$ of (4.3) with $|\mathcal{K}| < n - 3$. Let $\nu_i = |\mathtt{adj}(i, \mathcal{C}_{\mathcal{G}_n})|$ for all $i \in V_n$. From Lemma 5 in the Supplementary Materials, if $\gamma \to 0$, $C_1 \sim (\gamma l)^2/2 \to 0$. In contrast, C_2 is a constant. Therefore the term $\exp\{-C_2(n - \nu_i - \nu_j - 4)\}$ is negligible, and thus

$$\begin{aligned}\sup_{i,j,\mathcal{K}} \mathbb{P}\left(\left|\hat{z}_{i,j|\mathcal{K}} - z_{i,j|\mathcal{K}}\right| > \gamma\right) &\leq O(n - \nu_i - \nu_j)\exp\left\{-(\gamma l)^2(n - \nu_i - \nu_j - 4)/2\right\} \\ &\leq O(n - \nu_i - \nu_j)\exp\left\{-C_3(n - \nu_i - \nu_j)\gamma^2\right\},\end{aligned}$$

where C_3 is a constant.

Denote by $E_{i,j|\mathcal{K}}$ the event "an error occurred when testing partial correlation for zero at nodes i, j with conditional set $\mathcal{K}$". An error can be a type I error or a type II error, denoted by $E^I_{i,j|\mathcal{K}}$ and $E^{II}_{i,j|\mathcal{K}}$, respectively. Therefore $E_{i,j|\mathcal{K}} = E^I_{i,j|\mathcal{K}} \bigcup E^{II}_{i,j|\mathcal{K}}$, and

$$\begin{aligned}E^I_{i,j|\mathcal{K}} &: \sqrt{n - |\mathcal{K}| - 3}\left|\hat{z}_{i,j|\mathcal{K}}\right| > \Phi^{-1}(1 - \alpha/2) \text{ and } z_{i,j|\mathcal{K}} = 0, \\ E^{II}_{i,j|\mathcal{K}} &: \sqrt{n - |\mathcal{K}| - 3}\left|\hat{z}_{i,j|\mathcal{K}}\right| \leq \Phi^{-1}(1 - \alpha/2) \text{ and } z_{i,j|\mathcal{K}} \neq 0.\end{aligned}$$

Choose $\alpha = \alpha_n = 2(1 - \Phi(\sqrt{n}c_n/2))$, where c_n is defined in assumption (A3). Then

$$\begin{aligned}\sup_{i,j,\mathcal{K}} \mathbf{P}(E^I_{i,j|\mathcal{K}}) &= \sup_{i,j,\mathcal{K}} \mathbb{P}\left[\left|\hat{z}_{i,j|\mathcal{K}} - z_{i,j|\mathcal{K}}\right| > \sqrt{n/(n-|\mathcal{K}|-3)}c_n/2\right] \\ &\leq O(n-\nu_i-\nu_j)\exp\left[-C_4(n-\nu_i-\nu_j)c_n^2\right],\end{aligned}$$

for some constant C_4. With the same choice of α,

$$\begin{aligned}\sup_{i,j,\mathcal{K}} \mathbb{P}(E^{II}_{i,j|\mathcal{K}}) &= \sup_{i,j,\mathcal{K}} \mathbb{P}\left[\left|\hat{z}_{i,j|\mathcal{K}}\right| \leq \sqrt{n/(n-|\mathcal{K}|-3)}c_n/2\right] \\ &\leq \sup_{i,j,\mathcal{K}} \mathbb{P}\left[\left|\hat{z}_{i,j|\mathcal{K}} - z_{i,j|\mathcal{K}}\right| > c_n\left(1 - \sqrt{n/(n-|\mathcal{K}|-3)}/2\right)\right] \\ &\leq O(n-\nu_i-\nu_j)\exp\left[-C_5(n-\nu_i-\nu_j)c_n^2\right],\end{aligned}$$

for some constant C_5.

$$\begin{aligned}&\mathbb{P}(\text{an error occurs in the step 2 of PenPC algorithm}) \\ &\leq \sum_{(i,j)\in F_n} 2^{\nu_i+\nu_j} O((n-\nu_i-\nu_j))\exp\{-C_6(n-\nu_i-\nu_j)c_n^2\} \\ &\leq O\left[\sum_{i=1}^{p_n}\sum_{j\in \texttt{adj}(i,\mathcal{C}_{\mathcal{G}_n})} n2^{2q_n}\exp\left\{-C_6(n-2q_n)c_n^2\right\}\right] \\ &\leq O\left[np_nq_n\exp\left\{2q_n - C_6(n-2q_n)c_n^2\right\}\right] \\ &\leq O\left[np_nq_n\exp\left\{-C_6n^{1-2d_1} + C_7q_n\right\}\right] \qquad (5.20) \\ &\leq O\left[n^{b+1}\exp\left\{-C_6n^{1-2d_1} + n^a + C_7n^b\right\}\right]\end{aligned}$$

for a positive constant C_6 and C_7. This probability converges to zero as $n \to \infty$ when $0 < d_1 < \min\left(\frac{1-a}{2}, \frac{1-b}{2}\right)$. □

Proof of Corollary 2

Proof. From Corollary 1 and Theorem 3,

$$\begin{aligned}
&\mathbb{P}(\text{an error occurs in the PenPC algorithm}) \\
&= \mathbb{P}(\hat{\mathcal{C}}_{\mathcal{G}_n}(\boldsymbol{\theta}) \neq \mathcal{C}_{\mathcal{G}_n}) + \mathbb{P}(\hat{\mathcal{G}}_n^u(\alpha_n) \neq \mathcal{G}_n^u) \\
&= O\left(\exp\{2n^a - n^a \log(n)\}\right) + O\left(\exp\{-Cn^{1-2d_1}\}\right) \\
&= O\left(\exp\{-Cn^{1-2d_1}\}\right)
\end{aligned}$$

for $d_1 < min((1-a)/2, (1-b)/2)$. □

Bibliography

Anderson, T. (2003). *An Introduction to Multivariate Statistical Analysis. 2003*. Wiley, New York.

Andersson, S., Madigan, D., and Perlman, M. (1997). A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541.

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516.

Barabási, A. and Albert, R. (1999). Emergence of scaling in random networks. *science*, 286(5439):509–512.

Bareinboim, E. and Pearl, J. (2012). Causal inference by surrogate experiments: z-identifiability. *arXiv preprint arXiv:1210.4842*.

Bickel, P. and Levina, E. (2008). Regularized estimation of large covariance matrices. *The Annals of Statistics*, 36(1):199–227.

Bonn, S., Zinzen, R. P., Girardot, C., Gustafson, E. H., Perez-Gonzalez, A., Delhomme, N., Ghavi-Helm, Y., Wilczyński, B., Riddell, A., and Furlong, E. E. (2012). Tissue-specific analysis of chromatin state identifies temporal signatures of enhancer activity during embryonic development. *Nature genetics*, 44(2):148–156.

Cai, T. and Liu, W. (2011). Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association*, 106(494):672–684.

Cai, T. T., Li, H., Liu, W., and Xie, J. (2013a). Covariate-adjusted precision matrix estimation with an application in genetical genomics. *Biometrika*, 100(1):139–156.

Cai, X., Bazerque, J. A., and Giannakis, G. B. (2013b). Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations. *PLoS computational biology*, 9(5):e1003068.

Castelo, R. and Roverato, A. (2006). A robust procedure for gaussian graphical model search from microarray data with p larger than n. *The Journal of Machine Learning Research*, 7:2621–2650.

Chen, J. and Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771.

Chen, L. S., Emmert-Streib, F., Storey, J. D., et al. (2007). Harnessing naturally randomized transcription to infer regulatory relationships among genes. *Genome Biol*, 8(10):R219.

Chen, Y., Zhu, J., Lum, P. Y., Yang, X., Pinto, S., MacNeil, D. J., Zhang, C., Lamb, J., Edwards, S., Sieberts, S. K., et al. (2008). Variations in dna elucidate molecular networks that cause disease. *Nature*, 452(7186):429–435.

Chickering, D. (2002). Learning equivalence classes of bayesian-network structures. *The Journal of Machine Learning Research*, 2:445–498.

Chickering, D. (2003). Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554.

Clauset, A., Newman, M. E., and Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Colombo, D. and Maathuis, M. (2012). A modification of the pc algorithm yielding order-independent skeletons. *arXiv preprint arXiv:1211.3295*.

de Jong, S., Boks, M., Fuller, T., Strengman, E., Janson, E., de Kovel, C., Ori, A., Vi, N., Mulder, F., Blom, J., et al. (2012). A gene co-expression network in whole blood of schizophrenia patients is independent of antipsychotic-use and enriched for brain-expressed genes. *PloS one*, 7(6):e39498.

Dempster, A. (1972). Covariance selection. *Biometrics*, pages 157–175.

Dor, D. and Tarsi, M. (1992). A simple algorithm to construct a consistent extension of a partially oriented graph. *Technicial Report R-185, Cognitive Systems Laboratory, UCLA*.

Doss, S., Schadt, E. E., Drake, T. A., and Lusis, A. J. (2005). Cis-acting expression quantitative trait loci in mice. *Genome research*, 15(5):681–691.

Efron, B. (2004). Large-scale simultaneous hypothesis testing. *Journal of the American Statistical Association*, 99(465):96–104.

Efron, B. and Tibshirani, R. (2002). Empirical bayes methods and false discovery rates for microarrays. *Genetic epidemiology*, 23(1):70–86.

Erd6s, P. and Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61.

Erdős, P. and Rényi, A. (1960). On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61.

Fan, J., Feng, Y., and Wu, Y. (2009). Network exploration via the adaptive lasso and scad penalties. *The Annals of Applied Statistics*, 3(2):521.

Fan, J. and Lv, J. (2011). Nonconcave penalized likelihood with np-dimensionality. *Information Theory, IEEE Transactions on*, 57(8):5467–5484.

Foygel, R. and Drton, M. (2010). Extended bayesian information criteria for gaussian graphical models. *arXiv preprint arXiv:1011.6640*.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science's STKE*, 303(5659):799.

Hageman, R. S., Leduc, M. S., Korstanje, R., Paigen, B., and Churchill, G. A. (2011). A bayesian framework for inference of the genotype–phenotype map for segregating populations. *Genetics*, 187(4):1163–1170.

Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243.

Hoerl, A. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Højsgaard, S. and Lauritzen, S. (2008). Graphical gaussian models with edge and vertex symmetries. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):1005–1027.

Hotelling, H. (1953). New light on the correlation coefficient and its transforms. *Journal of the Royal Statistical Society. Series B (Methodological)*, 15(2):193–232.

Hughes, T., Marton, M., Jones, A., Roberts, C., Stoughton, R., Armour, C., Bennett, H., Coffey, E., Dai, H., He, Y., et al. (2000). Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126.

Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *The Journal of Machine Learning Research*, 8:613–636.

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M., and Bühlmann, P. (2012). Causal inference using graphical models with the r package pcalg. *Journal of Statistical Software*, 47(11):1–26.

Katan, M. B. (2004). Apolipoprotein e isoforms, serum cholesterol, and cancer. *International journal of epidemiology*, 33(1):9–9.

Kruglyak, L. and Storey, J. D. (2009). Cause and express. *Nature biotechnology*, 27(6):544–545.

Kulp, D. C. and Jagalur, M. (2006). Causal inference of regulator-target pairs by gene mapping of expression phenotypes. *BMC genomics*, 7(1):125.

Lam, C. and Fan, J. (2009). Sparsistency and rates of convergence in large covariance matrix estimation. *Annals of statistics*, 37(6B):4254.

Lauritzen, S. (1996). *Graphical models*, volume 17. Oxford University Press, USA.

Levina, E., Rothman, A., and Zhu, J. (2008). Sparse estimation of large covariance matrices via a nested lasso penalty. *The Annals of Applied Statistics*, pages 245–263.

Li, R., Tsaih, S.-W., Shockley, K., Stylianou, I. M., Wergedal, J., Paigen, B., and Churchill, G. A. (2006). Structural model analysis of multiple quantitative traits. *PLoS genetics*, 2(7):e114.

Li, Y., Tesson, B. M., Churchill, G. A., and Jansen, R. C. (2010). Critical reasoning on causal inference in genome-wide linkage and association studies. *Trends in Genetics*, 26(12):493–498.

Logsdon, B. A. and Mezey, J. (2010). Gene expression network reconstruction by convex feature selection when incorporating genetic perturbations. *PLoS computational biology*, 6(12):e1001014.

Lv, J. and Fan, Y. (2009). A unified approach to model selection and sparse recovery using regularized least squares. *The Annals of Statistics*, 37(6A):3498–3528.

Maathuis, M., Colombo, D., Kalisch, M., and Bühlmann, P. (2010). Predicting causal effects in large-scale systems from observational data. *Nature Methods*, 7(4):247–248.

Maathuis, M., Kalisch, M., and Bühlmann, P. (2009). Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164.

Magwene, P., Kim, J., et al. (2004). Estimating genomic coexpression networks using first-order conditional independence. *Genome Biol*, 5(12):R100.

Meek, C. (1995a). Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 403–410. Morgan Kaufmann Publishers Inc.

Meek, C. (1995b). Strong completeness and faithfulness in bayesian networks. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 411–418. Morgan Kaufmann Publishers Inc.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462.

Millstein, J., Zhang, B., Zhu, J., and Schadt, E. E. (2009). Disentangling molecular relationships with a causal inference test. *BMC genetics*, 10(1):23.

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003). Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 52(1):91–118.

Móri, T. (2005). The maximum degree of the barabási-albert random tree. *Combinatorics Probability and Computing*, 14(3):339–348.

Neto, E. C., Ferrara, C. T., Attie, A. D., and Yandell, B. S. (2008). Inferring causal phenotype networks from segregating populations. *Genetics*, 179(2):1089–1100.

Neto, E. C., Keller, M. P., Attie, A. D., and Yandell, B. S. (2010). Causal graphical models in systems genetics: a unified framework for joint inference of causal network and genetic architecture for correlated phenotypes. *The annals of applied statistics*, 4(1):320.

Pearl, J. (2000). *Causality: models, reasoning and inference*. Cambridge Univ Press.

Pearl, J. (2009). *Causality: models, reasoning and inference*. Cambridge Univ Press.

Pourahmadi, M. (2011). Covariance estimation: The glm and regularization perspectives. *Statistical Science*, 26(3):369–387.

Robins, J. M., Scheines, R., Spirtes, P., and Wasserman, L. (2003). Uniform consistency in causal inference. *Biometrika*, 90(3):491–515.

Rothman, A., Bickel, P., Levina, E., and Zhu, J. (2008). Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515.

Rothman, A., Levina, E., and Zhu, J. (2009). Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association*, 104(485):177–186.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D., and Nolan, G. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science's STKE*, 308(5721):523.

Schadt, E. E., Lamb, J., Yang, X., Zhu, J., Edwards, S., GuhaThakurta, D., Sieberts, S. K., Monks, S., Reitman, M., Zhang, C., et al. (2005). An integrative genomics approach to infer causal associations between gene expression and disease. *Nature genetics*, 37(7):710–717.

Schäfer, J. and Strimmer, K. (2005). An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764.

Schäfer, J., Strimmer, K., et al. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4(1):32.

Schmidt, M., Niculescu-Mizil, A., and Murphy, K. (2007). Learning graphical model structure using l1-regularization paths. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22,2, page 1278. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Schott, J. (2005). *Matrix analysis for statistics*. Wiley.

Sheehan, N., Didelez, V., Burton, P., and Tobin, M. (2008). Mendelian randomisation and causal inference in observational epidemiology. *PLoS medicine*, 5(8):e177.

Shojaie, A. and Michailidis, G. (2010). Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. *Biometrika*, 97(3):519–538.

Smith, G. (2007). Capitalizing on mendelian randomization to assess the effects of treatments. *JRSM*, 100(9):432–435.

Smith, G. and Ebrahim, S. (2003). mendelian randomization: can genetic epidemiology contribute to understanding environmental determinants of disease? *International Journal of Epidemiology*, 32(1):1–22.

Speed, T. and Kiiveri, H. (1986). Gaussian markov distributions over finite graphs. *The Annals of Statistics*, 14(1):138–150.

Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular biology of the cell*, 9(12):3273–3297.

Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, prediction and search*, volume 81. The MIT Press.

Stekhoven, D. J., Moraes, I., Sveinbjörnsson, G., Hennig, L., Maathuis, M. H., and Bühlmann, P. (2012). Causal stability ranking. *Bioinformatics*, 28(21):2819–2823.

Stuart, J., Segal, E., Koller, D., and Kim, S. (2003). A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–255.

Sun, W. (2012). A statistical framework for eqtl mapping using rna-seq data. *Biometrics*, 68(1):1–11.

Sun, W. and Hu, Y. (2013). eqtl mapping using rna-seq data. *Statistics in biosciences*, pages 1–22.

Sun, W., Yu, T., and Li, K.-C. (2007). Detection of eqtl modules mediated by activity levels of transcription factors. *Bioinformatics*, 23(17):2290–2297.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

Vignes, M., Vandel, J., Allouche, D., Ramadan-Alban, N., Cierco-Ayrolles, C., Schiex, T., Mangin, B., and de Givry, S. (2011). Gene regulatory network reconstruction using bayesian networks, the dantzig selector, the lasso and their meta-analysis. *PloS one*, 6(12):e29165.

Wille, A. and Bühlmann, P. (2006). Low-order conditional independence graphs for inferring genetic networks. *Statistical applications in genetics and molecular biology*, 5(1):1–32.

Wille, A., Zimmermann, P., Vranová, E., Fürholz, A., Laule, O., Bleuler, S., Hennig, L., Prelic, A., Von Rohr, P., Thiele, L., et al. (2004). Sparse graphical gaussian modeling of the isoprenoid gene network in arabidopsis thaliana. *Genome Biol*, 5(11):R92.

Yin, J. and Li, H. (2011). A sparse conditional gaussian graphical model for analysis of genetical genomics data. *The annals of applied statistics*, 5(4):2630.

Yin, J. and Li, H. (2013). Adjusting for high-dimensional covariates in sparse precision matrix estimation by 1-penalization. *Journal of Multivariate Analysis*, 116:365–381.

Yuan, M. (2010). High dimensional inverse covariance matrix estimation via linear programming. *The Journal of Machine Learning Research*, 99:2261–2286.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35.

Zhang, B., Horvath, S., et al. (2005). A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1):1128.

Zhang, C., Frias, M., Mele, A., Ruggiu, M., Eom, T., Marney, C., Wang, H., Licatalosi, D., Fak, J., and Darnell, R. (2010). Integrative modeling defines the nova splicing-regulatory network and its combinatorial controls. *Science's STKE*, 329(5990):439.

Zhou, S., Rütimann, P., Xu, M., and Bühlmann, P. (2011). High-dimensional covariance estimation based on gaussian graphical models. *The Journal of Machine Learning Research*, 999888:2975–3026.

Zhu, J., Wiener, M. C., Zhang, C., Fridman, A., Minch, E., Lum, P. Y., Sachs, J. R., and Schadt, E. E. (2007). Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations. *PLoS computational biology*, 3(4):e69.

Zhu, J., Zhang, B., Smith, E. N., Drees, B., Brem, R. B., Kruglyak, L., Bumgarner, R. E., and Schadt, E. E. (2008). Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nature genetics*, 40(7):854–861.