Bioinformatics, 31(24), 2015, 3953–3960
doi: 10.1093/bioinformatics/btv513
Advance Access Publication Date: 2 September 2015
Original Paper

OXFORD

Genetics and population analysis

Learning directed acyclic graphical structures with genetical genomics data

Bin Gao and Yuehua Cui*

Department of Statistics and Probability, Michigan State University, East Lansing, MI 48824, USA

*To whom correspondence should be addressed.

Associate Editor: Gunnar Ratsch

Received on January 7, 2015; revised on July 20, 2015; accepted on August 24, 2015

Abstract

Motivation: Large amount of research efforts have been focused on estimating gene networks based on gene expression data to understand the functional basis of a living organism. Such networks are often obtained by considering pairwise correlations between genes, thus may not reflect the true connectivity between genes. By treating gene expressions as quantitative traits while considering genetic markers, genetical genomics analysis has shown its power in enhancing the understanding of gene regulations. Previous works have shown the improved performance on estimating the undirected network graphical structure by incorporating genetic markers as covariates. Knowing that gene expressions are often due to directed regulations, it is more meaningful to estimate the directed graphical network.

Results: In this article, we introduce a covariate-adjusted Gaussian graphical model to estimate the Markov equivalence class of the directed acyclic graphs (DAGs) in a genetical genomics analysis framework. We develop a two-stage estimation procedure to first estimate the regression coefficient matrix by ℓ_1 penalization. The estimated coefficient matrix is then used to estimate the mean values in our multi-response Gaussian model to estimate the regulatory networks of gene expressions using PC-algorithm. The estimation consistency for high dimensional sparse DAGs is established. Simulations are conducted to demonstrate our theoretical results. The method is applied to a human Alzheimer's disease dataset in which differential DAGs are identified between cases and controls. R code for implementing the method can be downloaded at http://www.stt.msu.edu/ ~cui.

Availability and implementation: R code for implementing the method is freely available at http://www.stt.msu.edu/~cui/software.html

Contact: cui@stt.msu.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The past decades have witnessed enormous methodology developments in gene network inference using gene expression data (e.g. Cheung and Spielman, 2002; Schadt *et al.*, 2003). Such networks were developed mostly by assessing the pairwise correlation of gene expressions. From a statistical point of view, under the assumption that the joint distribution of the gene expressions of interest is a multivariate normal distribution, such networks can be constructed by assessing the nonzero elements of the inverse covariance matrix, the so called precision matrix or concentration matrix. Assume $X = (X_1, ..., X_p) \sim N(0, \Sigma)$ and $\Omega = \Sigma^{-1}$. $X_i \perp\!\!\!\perp X_j | \{X_t\}_{t \neq i,j} \Leftrightarrow \omega_{i,j} = 0$, where $\omega_{i,j}$ is the (i, j)th entry in Ω. That is, X_i and X_j are conditionally independent if and only if $\omega_{i,j} = 0$. Thus, one can infer the positions of zero entries in Ω to infer the graphical structure, i.e. the conditional independencies between components in X. However, when the data dimension is larger than the sample size, the inverse of the sample covariance matrix is not available. When considering such networks as undirected graphs, there has been huge statistical interests in developing Gaussian graphical models assuming sparsity of the precision matrix, to name a few, Friedman *et al.* (2008), Cai *et al.* (2013), Meinshausen and Bühlmann (2006) and Yuan

and Lin (2007). Under certain assumptions, the positions of zero entries in the precision matrix indicate conditional independence between variables of study.

When inferring a network only based on gene expressions, two genes could show correlation in expression simply because they share a common regulator, while they should be independent conditioning on the common regulator. Motivated by this, a few methodological developments have been focused on covariate-adjusted graphical models by using genetic markers as covariates to correct both false positives and false negatives (e.g. Cai *et al.*, 2013; Lee and Liu, 2012; Yin and Li, 2011, 2013). In their estimation procedures, the effect of genetic variants is estimated in the first step. Then, the graphical structure is estimated in the second step while adjusting for the genetic effects. Their simulation results and real data analysis showed that graph estimates were substantially improved when taking the genetic effects into account. These models are all focused on the estimation of undirected graphs without considering direction information. The directed graphs, however, are more attractive in the sense that often people are interested in the causal relationships, i.e. interests are not only focused on whether two variables are conditionally independent but also if one causes the other one. In real life, gene expressions are often the results of directed regulations. Thus, the estimation of directed graphs should provide more biological information for further functional investigation. Genetical genomics data provide optimal resources to learn such directed graphical networks.

In typical directed acyclic graph (DAG) inference, one probability distribution has a set of corresponding DAGs which are Markov equivalent under certain assumptions. The directions of the edges in the graphs indicate causal relations. Studies on directed graphs have flourished in the literature both in theory and in application (e.g. Ali *et al.*, 2009; Anderson *et al.*, 1997; Richardson and Spirtes, 2002). The most common case is the Gaussian case. Let $Y = (Y_1, ..., Y_p)$ be a p-dimensional Gaussian random vector, and each coordinate represents a gene expression. To infer the regulatory relationship of these p gene expressions, the PC-algorithm developed by Spirtes *et al.* (2000) can be applied to construct a completed partially DAG (CPDAG). The CPDAG can uniquely represent all DAGs corresponding to the common distribution of Y and can be represented as the regulatory networks of the gene expressions. Kalisch and Bühlmann (2007) later proved that the graph obtained from the PC-algorithm is consistent under some conditions, even in high dimensional cases.

Directed graphs constructed with the PC-algorithm are more biologically relevant compared with undirected graphs. However, as mentioned before, if two genes share a common regulator, their expression relationship could be complicated or even distorted without considering the underlying genetic structure. When adjusting for the genotype (i.e. the common regulator) effects, the two related genes could be independent or vice versa. This motivates us to develop a covariate-adjusted directed graphical model by extending the CPDAG estimation procedure to a regression framework. We treat genetic markers as covariates and develop a two-stage estimation method, which combines the penalized estimation and PC-algorithm to estimate the marker-adjusted CPDAG structure.

In the following of the article, we first provide some background knowledge about graphical models and the formulation of our model and the two-stage estimation procedure in Section 2. Theoretical results are given in Section 3. Simulations and real data analysis are given in Sections 4 and 5 to justify our method followed by discussions in Section 6. All the proofs of the theoretical results are rendered in the Supplementary File due to space limit.

2 Model and estimation

2.1 Background

Let $G = (V, E)$ be a graph where V represents the set of vertices and E represents the set of edges. A directed graph is a graph in which each edge has a direction represented by a " $\rightarrow$" sign. A DAG is a directed graph with no directed cycles. A criterion called d-separation is used to read conditional independence relationships from a DAG structure (Pearl, 2000). Other criteria can be used to generate conditional independence relationships. The reason why d-separation is particularly useful is that it is related to causal inference (Pearl, 2009; Spirtes *et al.*, 2000) and for the purpose of constructing directed graphs.

Given a random vector, its coordinates can be viewed as the vertices of a graph $G = (V, E)$. A probability distribution is called faithful to a DAG if and only if the conditional independence relationships indicated by the distribution are the same as those obtained via d-separation. Multiple DAGs may represent the same set of conditional independence relationships. The DAGs to which a probability distribution is faithful are Markov equivalent and compose a Markov equivalence class. Two DAGs are Markov equivalent if and only if they have the same skeleton and the same v-structures (Frydenberg, 1990; Verma and Pearl, 1990, 1992). The skeleton of a DAG is the graph where all directed edges are replaced with undirected edges. A v-structure is a triple (v_1,v_2,v_3) having the structure $v_1 \rightarrow v_2 \leftarrow v_3$.

A Markov equivalence class of DAGs can be described by a unique CPDAG (Chichering, 2002). A CPDAG is the union of the DAGs in a Markov equivalence class in the sense that a directed edge exists if that edge exists in each DAG and an undirected edge exists if that edge has different directions in two DAGs. PC-algorithm (Spirtes *et al.*, 2000) is an effective and efficient algorithm to estimate CPDAGs. The reason why we estimate the CPDAG is that multiple DAGs correspond to the same probability distribution, while only one CPDAG corresponds to a probability distribution. After a CPDAG is obtained, we can extend it to the DAGs by adding directions on the edges, such that no directed cycles are generated. It has been proven that PC-algorithm can generate a consistent estimate under certain assumptions, even in high dimensional cases (Kalisch and Bühlmann, 2007).

2.2 Covariate-adjusted Gaussian graphical model

Consider a multi-response linear regression model with p responses and q covariates. Suppose we have n independent observations (y_i, x_i), $i = 1, ..., n$, where $y_i = (y_{i1}, ..., y_{ip})^T$ and $x_i = (x_{i1}, ..., x_{iq})^T$. We assume that the data are centered and standardized. Define $Y = (y_1, ..., y_n)^T$ and $X = (x_1, ..., x_n)^T$. The relationship between the multivariate response Y and the covariates X can be described by the following regression model,

$$Y = XB + E$$

where $B = (B_1, ..., B_p)$ is a $q \times p$ coefficient matrix and $E = (\epsilon_1, ..., \epsilon_n)^T$ is the error term. We assume that the errors $\epsilon_i = (\epsilon_{i1}, ..., \epsilon_{ip})^T$, $i = 1, ..., n$, are *i.i.d.* random variables following a multivariate normal distribution $N(0, \Sigma)$. Let $y^j = (y_{1j}, \cdots, y_{nj})^T$ be the jth response vector ($j = 1, .., p$) and $x^l = (x_{1l}, \cdots, x_{nl})^T$ be the lth covariate vector ($l = 1, .., q$). In a genetical genomics framework, X represents the single-nucleotide polymorphism (SNP) markers and Y represents the gene expressions. Our interest is to estimate the CPDAG structure of Y, while adjusting for the effect of X. Conditional on X, we expect that false positives and/or false negatives of connections will be reduced.

2.3 A two-stage estimation procedure

Without X, the PC-algorithm can be applied to infer the CPDAG structures. In the model described above, the conditional mean of Y changes as X varies. Thus, the PC-algorithm cannot be directly applied since it is only for a model with a constant mean, i.e. $X = (X_1, ..., X_q)$ follows a multivariate normal distribution with mean μ and covariance matrix Σ. To overcome the difficulty, we propose a two-stage estimation procedure to estimate the mean function and the CPDAG structure. In the first stage, we estimate the coefficient matrix B. We implement a penalized estimation procedure with respect to each response in Y via solving the following p optimization problems,

$$\hat{B}_j = \arg\min_{B_j} \left[\frac{1}{2n} tr\left\{ (y^j - XB_j)^T (y^j - XB_j) \right\} + \lambda_{j,n} ||B_j||_1 \right],$$

where $\lambda_{j,n}, j = 1, ..., p$, are tuning parameters, and $|| \cdot ||_1$ refers to the L_1 norm. The estimate of B is denoted as $\hat{B} = (\hat{B}_1, ..., \hat{B}_p)$. Let $\hat{E} = Y - X\hat{B}$. In the second stage, the PC-algorithm is applied to $\hat{E}$ to estimate the directed graphical structure corresponding to the probability model $N(0, \Sigma)$. The two-stage estimation algorithm is termed as the covariate-adjusted PC-algorithm (caPC).

The caPC contains two steps. The first step is to estimate the skeleton, and the second step is to extend the skeleton to the CPDAG. Information obtained from the first step is used to determine the directions of the edges in the second step. If we have the perfect knowledge about the conditional independence relationships, the PC-algorithm can return us with the correct skeleton in the first step (Spirtes *et al.*, 2000). However, we do not have perfect knowledge in practice. Therefore, we use samples to estimate the skeleton by implementing a testing procedure. For testing the conditional independence, Kalisch and Bühlmann (2007) applied estimated partial correlations and Fisher transformation to construct a testing statistic. They proved that the PC-algorithm could generate consistent estimates under certain assumptions even in the high dimensional situations. To make the article self-contained, we also included the PC algorithm in the Supplementary File.

3 Theoretical result

In this work, we allow the number of responses to increase along with the sample size, i.e. p is a function of n. The number of marker covariates q is fixed. This is a valid treatment as we can do an eQTL mapping study to first find the eQTLs for the expression responses, then fix them in the model. Below we show that the two-stage estimation method generates consistent estimate of a CPDAG. To show the estimation consistency, we adopt the setup of Knight and Fu (2000) and Kalisch and Bühlmann (2007). The dimension of a multivariate response variable is denoted as $p(n)$ and a DAG is denoted as $G = G_n$.

Let $\rho_{n;i,j}$ be the correlation between ϵ_i and ϵ_j. Let $\rho_{n;i,j|\mathbf{k}}$ be the partial correlation between ϵ_i and ϵ_j given $\{\epsilon_r, r \in \mathbf{k}\}$, where $\mathbf{k}$ is a subset of $\{1, ..., p(n)\} \backslash \{i, j\}$. $\hat{\rho}_{n;i,j}$ and $\hat{\rho}_{n;i,j|\mathbf{k}}$ are the corresponding estimated ones.

To establish the estimation consistency, we need the following lemmas. The required conditions for the lemmas and theorem can be found in the Supplementary File.

Lemma 1: *Assume that the distribution P_n is normally distributed, we have*

$$\hat{\rho}_{n;i,j}(\hat{B}) - \hat{\rho}_{n;i,j}(B) = o_p(1), \forall\ i, j = 1, ..., p.$$

Lemma 2: *Assume that the distribution P_n is normally distributed, we have*

$$\hat{\rho}_{n;i,j|\mathbf{k}}(\hat{B}) - \hat{\rho}_{n;i,j|\mathbf{k}}(B) = o_p(1), \forall\ i, j = 1, ..., p.$$

Lemma 3: *Assume that the distribution P_n is normally distributed and $\sup_{n,i,j,\mathbf{k}} |\rho_{n;i,j|\mathbf{k}}| \leq M < 1$. Then, for any $\gamma > 0$, we have*

$$\sup_{i,j,\mathbf{k} \in K_{ij}^{m_n}} \mathbf{P}\left(|\hat{\rho}_{n;i,j|\mathbf{k}}\left(\hat{B}\right) - \rho_{n;i,j|\mathbf{k}}| > \gamma \right) \leq C_1 (n - 2 - m_n) * \exp\left((n - 4 - m_n) \log\left(\frac{4 - \gamma^2}{4 + \gamma^2} \right) \right)$$

for some positive constant C_1.

Lemma 1 shows that the difference between the estimated correlation coefficient $\hat{\rho}_{n;i,j}$ given the estimated coefficient matrix $\hat{B}$ and the estimated correlation coefficient given the true coefficient matrix B is $o_p(1)$. Lemma 2 states that the difference between the estimated partial correlation coefficient $\hat{\rho}_{n;i,j|\mathbf{k}}$ given the estimated coefficient matrix $\hat{B}$ and the estimated partial correlation coefficient given the true coefficient matrix B is $o_p(1)$. Lemma 3 shows that the difference between the estimated partial correlation coefficient $\hat{\rho}_{n;i,j|\mathbf{k}}$ given the estimated coefficient matrix $\hat{B}$ and the true partial correlation coefficient $\rho_{n;i,j|\mathbf{k}}$ is bounded. The proofs of the three lemmas are relegated in the Supplementary File.

Define α_n as the significance level for testing the significance of the partial correlation after Fisher's transformation, i.e. for testing $H_0 : \rho_{i,j|k} = 0$. With the above assumptions, we have the following theorem.

Theorem: *Define $\hat{G}(\alpha_n)$ as the estimated CPDAG using the two-stage estimation procedure and G is the true CPDAG. If $\lambda_{j,n} = o(n)$, $j = 1, ..., p$ and Conditions (C1)-(C6) are satisfied, then there exists α_n such that*

$$\mathbf{P}(\hat{G}(\alpha_n) = G) = 1 - O(\exp(-cn^{1-2d})) \to 1, \text{as } n \to \infty$$

where c is a positive constant and $d > 0$ is as in Condition (C6).

The convergent rate of the CPDAG estimation is the same as that in Theorem 2 in Kalisch and Bühlmann (2007). This means that the errors occurred in the first stage do not have a great effect on the second stage of the estimation of graphical structures. The proof of the Theorem can be found in the Supplementary File.

4 Simulation

4.1 Simulation design

We did extensive simulations to assess the estimation performance. We followed the simulation setup in Yin and Li (2013) and Kalisch and Bühlmann (2007). To generate the error term, we began with an adjacency matrix $A_{p \times p}$ filled with zeros. Then, every entry in the lower triangle was replaced with independent realizations of a Bernoulli random variable with success probability p_A $(0 < p_A < 1)$, where p_A is called the sparseness of the model. Entries with 1 were then replaced by realizations of a Uniform (0,1) random variable. The corresponding DAG was constructed by drawing a directed edge from node s to node t if $A_{st} \neq 0$ and $s < t$. The created DAG has the following property: $\mathbf{E}(N_j) = p_A(p - 1)$, where N_j is the number of neighbors for a given node j. We denote E[N] as the average number of neighbors per node. The size of $\mathbf{E}(N_j)$ determines the sparseness of the graph with small values corresponding to a sparse graph and large values corresponding to a dense graph.

The matrix A will be used to generate the data as follows.

$$\eta_j \sim N(0,1), j = 1, \cdots, p,$$
$$\epsilon_1 = \eta_1,$$
$$\epsilon_j = \sum_{k=1}^{j-1} A_{jk}\epsilon_k + \eta_j,$$

where $\eta_j's$ are independent. We applied the R-package *pcalg* to generate DAGs according to the above description. Such processes were repeated to generate the error matrix $E_{n\times p}$.

To generate the $q \times p$ coefficient matrix B, we generated a $q \times p$ sparse indicator matrix whose entry is 1 with a probability proportional to κ/q, where κ is called the sparseness of the coefficient matrix. If an entry is 1, it is replaced by a realization from Unif([−1,−u]∪[u,1]), where u is a realization from Unif([0.1, 0.9]). Next we generated X whose entries were realizations of Bernoulli (1, 0.5). Finally, we generated Y by $Y = XB + E$.

The performance was evaluated based on true-positive rate (TPR), false-positive rate (FPR) and the structural Hamming distance (SHD) following the evaluation criteria proposed by Kalisch and Bühlmann (2007). TPR and FPR were used for the evaluation of the estimate of the skeleton. TPR is defined as the ratio of the number of correctly estimated edges to the number of total edges. FPR is defined as the ratio of the number of incorrectly estimated edges to the number of total nonedges (the number of locations where there are no edges). SHD was used for the evaluation of the estimated CPDAG and is defined as the number of edges where the true CPDAG and the estimated CPDAG are different.

4.2 Performance in the low dimensional case

We chose $\alpha_n = 0.01$ in the low dimensional situation where p was fixed and n varied. We called it low dimensional case in which the sample size n is larger than p and q. For comparison purpose, we also included the $\log(n) = 4$ case. Three parameter settings and seven sample sizes were chosen, each with 40 replications. Figure 1 depicts the simulation results. In all the cases, TPR increases and SHD decreases as the sample size increases (denoted by $\log(n)$ in the plot). The tendency of FPR is not clear. However, the overall scale of FPR is small in all cases, implying reasonable control of FPR. A similar pattern was also observed in Kalisch and Bühlmann (2007). The authors explained that this is because a constant α_n was used under all sample sizes. In addition, the performance of a sparse graph estimation (denoted as circles with $\mathbf{E}[N] = 2$) appears to be better than that of a dense graph estimation (denoted as triangles with $\mathbf{E}[N] = 5$). The 95% confidence interval based on the 40 replicates for each case is also shown. It is clear that the confidence interval becomes narrower as the sample size increases in all the cases. This is consistent with our large sample theoretical result.

4.3 Performance in the high dimensional case

In this section, we reported the performance of the estimation in high dimensional situations where p varied as n increased, to check the consistency results as stated in Theorem 1. In all the cases, p was assumed to be larger than n. We closely followed the setup in Kalisch and Bühlmann (2007) and chose $\alpha_n = 0.05$ in the high dimensional case. Specifically, data were simulated following Table 1, and the number of covariates was assumed to be $q = 100$ and 500. The average number of neighbors was calculated as $\mathbf{E}[N] = n^{0.5}/6$. The sparseness of the coefficient matrix was controlled by κ. We ran 30 simulations in each setting. Figures 2 and 3 show the results for $q = 100$ and $q = 500$, respectively. As shown in both figures, it is clear that TPR is increasing and FPR is decreasing along with the increasing of the sample size. These observations are consistent with our theoretical results. In addition, the size of q also has an effect on the estimation performance in which larger q (=500) gives smaller TPR and larger FPR compared with the results under smaller q (=100).

We did another simulation in which both p and q increase with n. The sample size n was assumed to be 50, 100, 150, 200, 250 and 300. The dimension of Y and X was assumed to be $p = n^{1.5}/6.3$ and $q = p/2$, respectively. The average number of neighbors is given as $\mathbf{E}[N] = n^{0.5}/6$. The detailed setup can be found in Table 2. Thirty simulations were run in each setting. The results of the simulation are shown in Figure 4. We observed a similar pattern as shown in Figures 2 and 3, i.e. TPR is increasing and FPR is decreasing as n increases. This result provides supportive evidence that our method also works in the situation where both p and q increase with n.

4.4 Comparing graphs with or without covariates

The proposed method improves the DAG estimation by adjusting for potential marker effects. To evaluate the gain by adjusting for the marker effects, we compared the estimation performance of before and after adjusting for the marker effects using the PC-algorithm (denoted as PC in the table) and caPC based on simulated data. We considered six scenarios as listed in Table 3. We observed consistently larger TPR, smaller FPR and SHD by using the caPC estimation method compared with the results using the PC-algorithm in all the simulation scenarios. This simulation further justifies the benefit of adjusting for the covariates' effect when estimating the DAG structure. We also did simulations considering the case where $q > p$ following the suggestion of one reviewer. The results are summarized in Table 4. We observed similar pattern in which caPC outperformed PC.

4.5 Comparing directed and undirected graphs

We did additional simulations to compare the performance of the directed and undirected graph estimation adjusting for the covariates' effects. Here, we choose $p = 50$, 100, 200, $q = 2p$ and $n = 250$. As before, we chose $E[N] = 2$, $\kappa = 3.5$. The results are summarized in Table 5. We applied graphical LASSO to estimate the undirected graphical structure (Friedman *et al.* 2008). We can see that undirected method achieves a higher TPR, but the directed method achieves a lower FPR and smaller SHD. Here, SHD is used to measure the skeleton of the graph since undirected graphs do not have direction information. The reason that undirected method is better on TPR is because, in theory, directed graphs give sparser solutions than the undirected method due to the extra constraint on direction inference. So overall, the directed graph performs better than the undirected method on both FPR and SHD but suffers a little bit on TPR.

5 Application to real data

We applied our proposed two-stage estimation method to an Alzheimer disease dataset (Webster *et al.* 2009). Following the article, 176 Alzheimer cases and 187 controls were included in our analysis. For the SNPs, those with low genotyping call rate (<90%), low minor allele frequency (<5%) and those that failed the Hardy–Weinberg equilibrium test in control group (P value < 0.001) were removed. After these operations, around 332 000 SNP markers were left for further analysis. We used the residual expression data after adjusting for the effects of several covariates such as gender, APOE

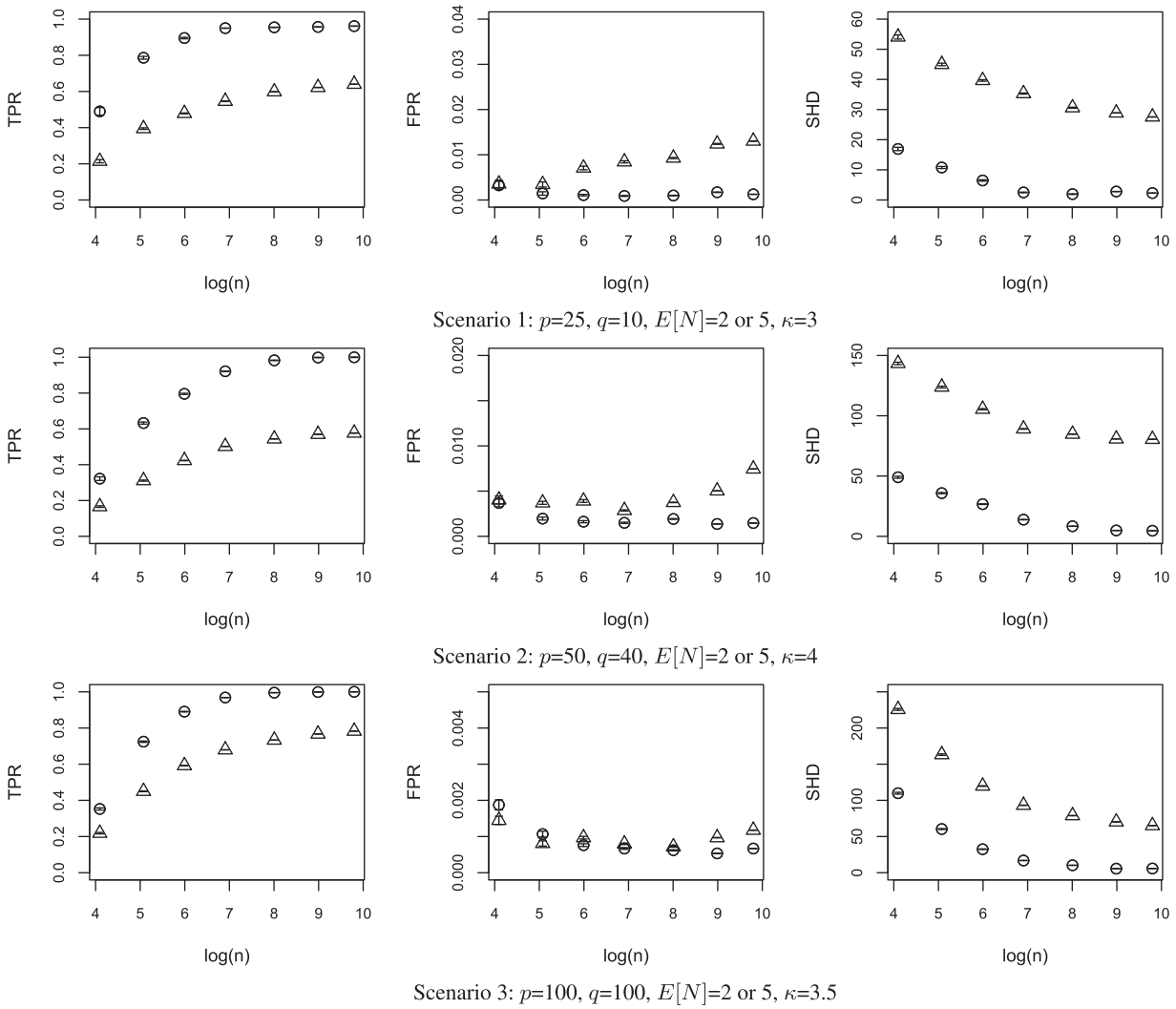

Fig. 1. Performance of the two-stage estimation algorithm under different parameter and sample size settings. The circles represent the sparse graph estimation results with $\mathrm{E}[N] = 2$ and the triangles represent the dense graph estimation results with $\mathrm{E}[N] = 5$. The upper and lower bounds of the 95% confidence intervals are also shown which are difficult to separate under large n

Table 1. Simulation setup in the high dimensional case with fixed κ

	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
p	162	398	674	980	1310	1661
n	50	100	150	200	250	300
$E[N]$	1.18	1.67	2.04	2.36	2.64	2.89
κ	5	5	5	5	5	5

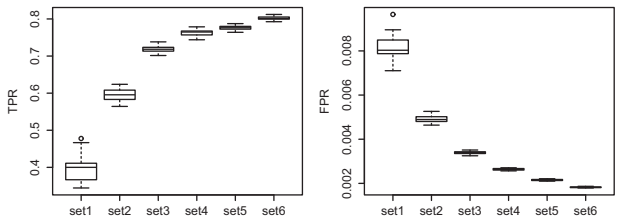

Fig. 2. Boxplot of TPR and FPR under different settings when $q = 100$

status and age at death (data can be found at http://labs.med.miami.edu/myers/LFuN/data ajhg.html). There are 8560 gene expressions. We implemented a two sample t-test for each gene expression and selected top 100 significant gene expressions to learn their DAG structure. For each of the 100 gene expressions, we fitted a simple linear regression with each SNP marker as the covariate and selected SNPs with P value < 0.001. Among these SNPs, we chose those that show association with two or more gene expressions (P value < 0.001). This ended up with 3776 SNPs in the case group and 3871 SNPs in the control group with 60 SNPs in common.

The final dataset used for the analysis contained 100 gene expressions (as the Y variable), 3776 SNPs in the case group and 3871 SNPs in the control group (as the X variables). We applied our estimating method to the two groups separately. Our goal was to learn the DAG structure for the 100 genes while adjusting for the effects of SNP markers in the case and control groups with the hope to identify differential DAG structures that can distinguish the two groups. Any differences might potentially explain the disease etiology of Alzheimer.

We learned the CPDAGs at the $\alpha_n = 0.001$ level. For the case group, the estimated CPDAG has 101 directed edges (Fig. 5). For the control group, the estimated CPDAG has 88 directed edges

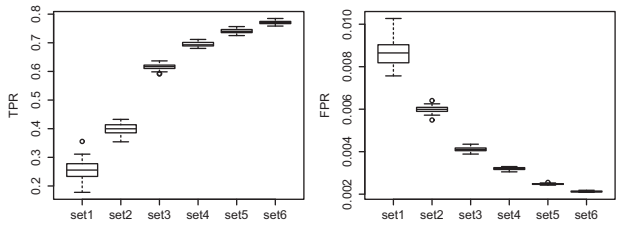

Fig. 3. Boxplot of TPR and FPR under different settings when $q = 500$

Table 2. Simulation setup in the high dimensional case with varying p, q and κ

	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
p	71	200	367	566	791	1039
q	35	100	184	283	395	520
n	50	100	150	200	250	300
$E[N]$	1.18	1.67	2.04	2.36	2.64	2.89
κ	3.9	4.6	5.0	5.3	5.5	5.7

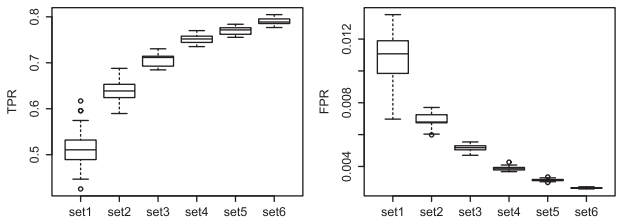

Fig. 4. Boxplot of TPR and FPR under different simulation settings with varying q and κ

Table 3. Estimation comparison with and without adjusting for the marker effects under different scenarios

Simulation setup	Method	TPR	FPR	SHD
$p = 25$, $q = 10$, $n = 250$,	caPC	0.768	0.001	11.5
$E[N] = 2$, $\kappa = 3.5$	PC	0.670	0.038	25.9
$p = 50$, $q = 50$, $n = 250$,	caPC	0.746	0.002	26.7
$E[N] = 2$, $\kappa = 4$	PC	0.568	0.020	55.3
$p = 100$, $q = 100$, $n = 250$,	caPC	0.740	0.002	52.4
$E[N] = 2$, $\kappa = 3$	PC	0.680	0.006	81.7
$p = 400$, $q = 200$, $n = 250$,	caPC	0.768	0.005	648.6
$E[N] = 2.5$, $\kappa = 20$	PC	0.235	0.014	1504.9
$p = 800$, $q = 200$, $n = 250$,	caPC	0.782	0.005	1948.1
$E[N] = 1.5$, $\kappa = 25$	PC	0.138	0.008	3175.9
$p = 1000$, $q = 200$, $n = 250$,	caPC	0.790	0.004	2543.4
$E[N] = 1.5$, $\kappa = 20$	PC	0.274	0.007	3874.4

(Fig. 5). The graphs were generated with the R package *igraph*. In both graphs, genes were represented with numbers. The corresponding gene names can be found in the Supplementary Table. The two CPDAGs share 18 edges in common. We noticed that some sub-CPDAG structures shown in the case group were not present in the control group. For example, the graph $70 \rightarrow 95 \leftarrow 38$ was observed in the case group, while the three genes are separated from other groups in the control population. On the other hand, some CPDAGs shown in the control group were not present in the case group. For example, the sub-graph by 17-26-31-32-76 is observed in the control group, while the genes are less connected in the case group.

Table 4. Estimation comparison between PC and caPC when $q > p$

Simulation setup	Method	TPR	FPR	SHD
$p = 100$, $q = 200$, $n = 200$,	caPC	0.818	0.009	84.0
$E[N] = 2$, $\kappa = 3$	PC	0.726	0.016	129.8
$p = 200$, $q = 400$, $n = 200$,	caPC	0.661	0.011	324.9
$E[N] = 2$, $\kappa = 20$	PC	0.265	0.018	516.4
$p = 400$, $q = 800$, $n = 200$,	caPC	0.570	0.008	866.7
$E[N] = 2$, $\kappa = 20$	PC	0.298	0.010	1138.3

Table 5. Estimation comparison between directed and undirected graphs

	Method	TPR	FPR	SHD
$p = 50$, $q = 100$, $n = 250$,	Directed	0.790	0.001	12.5
$E[N] = 2$, $\kappa = 3.5$	Undirected	0.874	0.037	49.9
$p = 100$, $q = 200$, $n = 250$,	Directed	0.764	0.001	31.9
$E[N] = 2$, $\kappa = 3.5$	Undirected	0.833	0.015	90.2
$p = 200$, $q = 400$, $n = 250$,	Directed	0.782	0.001	75.1
$E[N] = 2.5$, $\kappa = 3.5$	Undirected	0.812	0.005	137.5

Such differential graphical structures between the two groups might indicate regulation heterogeneity between the two groups. Further biological verifications are needed to validate the findings.

As a comparison, we applied PC-algorithm directly to the 100 genes without adjusting for the SNPs' effects. For the case group, the estimated graph has 124 edges. For the control group, the estimated graph has 100 edges. This is consistent with our assumption that the number of edges should be reduced after adjusting for the markers' effect. This phenomenon has been explained and demonstrated in other works focusing on undirected graphs (e.g. Cai *et al.* 2013; Yin and Li, 2011).

We did another case study by focusing on the Alzheimer's disease pathway from the KEGG database. There are total 168 genes in this pathway, but only 120 gene expressions in this dataset were mapped to the pathway. For each of the 120 gene expressions, we performed a single marker linear regression analysis using each of the SNPs. We selected those markers with P value < 0.001. Then the 5000 SNPs (with 93 in common) with the smallest P values were selected in case group and control group, respectively.

The data we used for the final analysis contain 120 gene expressions (the Y variables), 5000 SNP markers in the case group and 5000 SNP markers in the control group (the X variables). Our goal is to learn the DAG structures based on the 120 gene expressions while adjusting for the effects of 5000 SNP markers in both groups. We applied our estimating method to the case and control group separately. In the case group, the estimated CPDAG has 108 directed edges (Fig. 6). In the control group, the estimated CPDAG has 109 directed edges (Fig. 6). There are 25 common edges between the two graphs. We can see different graphical structures between the two groups. Some of the estimated sub-DAGs agree with the real biological network. For example, some genes in mitochondria were identified as connected in the case graph (e.g. COX7A2, NDUFB2, SDHC; NDUFV3, COX7B, COX4I1, COX1, NDUFB4, NDUFA10).

6 Discussion

Genes function in networks and gene expressions often result from gene regulations. Hence, learning directed regulations can enhance

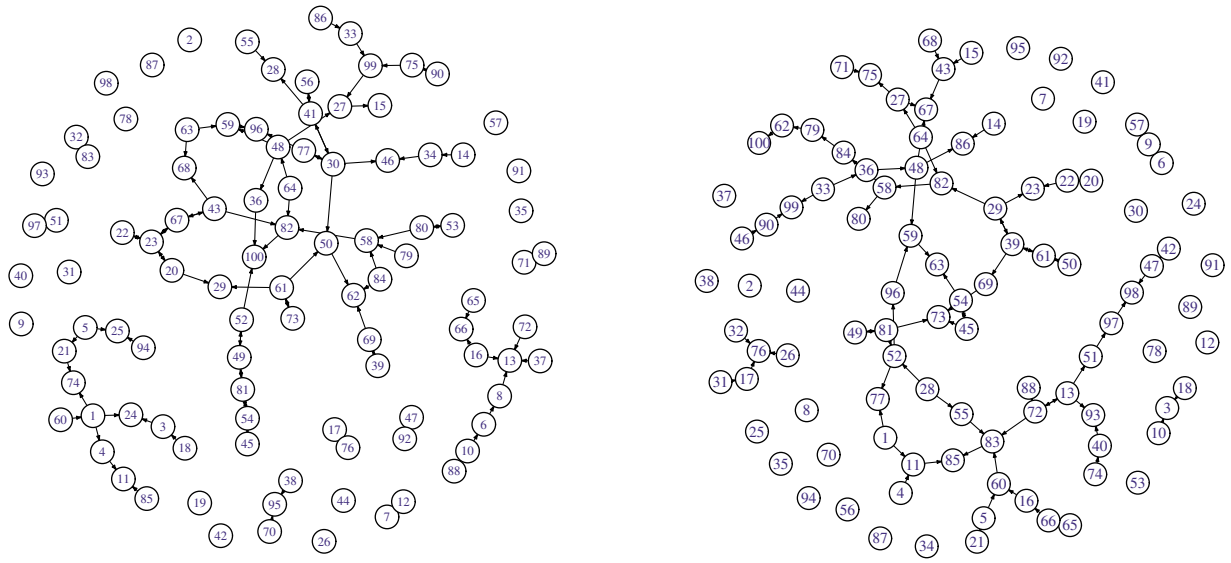

Fig. 5. The estimated CPDAG in the case (left) and control (right) group based on the top 100 genes. Those overlapped nodes indicate dual regulation directions

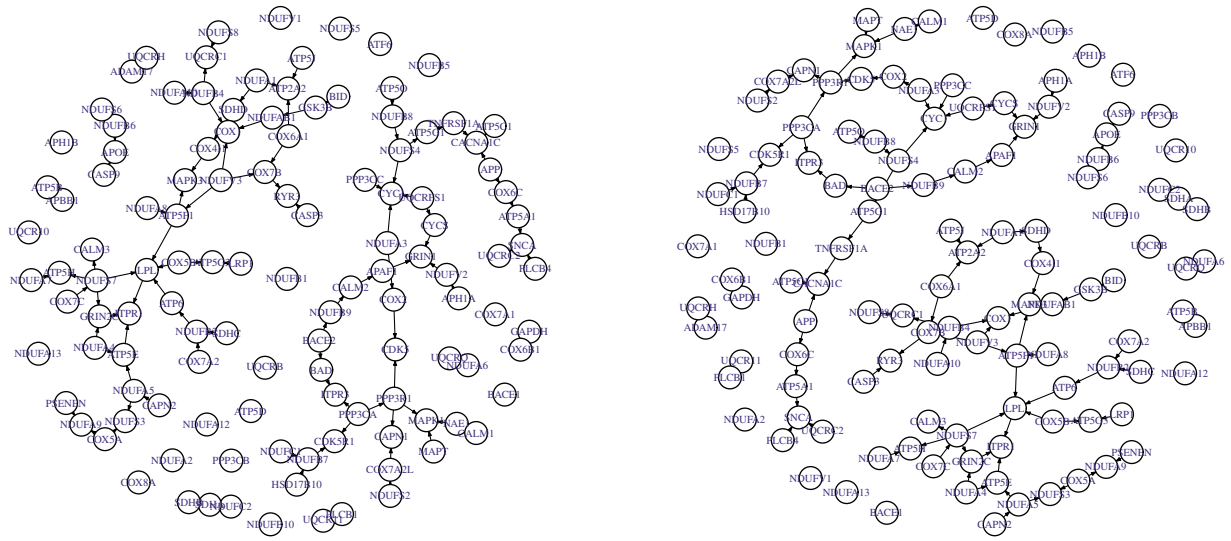

Fig. 6. The estimated CPDAG in the case (left) and control (right) group based on genes mapped to the Alzheimer's disease pathway from the KEGG database. Those overlapped nodes indicate dual regulation directions

our knowledge about gene function and disease etiology. In this work, we introduced a covariate-adjusted model to study the graphical structures of multiple gene expressions with direction regulation information. We proposed a two-stage procedure to obtain the estimated CPDAG corresponding to the probability distribution. The marker adjusted expression network can reduce the false connections of two genes if they share a common regulator. Simulated data were used to test our method and showed supportive results both in low and high dimensional cases. We provided a consistency result for the case when p is increasing along with sample size n. We applied our method to a real case-control dataset to understand the gene regulation mechanism in Alzheimer disease. Different graphs were learned corresponding to the case and control group. For the Alzheimer pathway analysis, we were able to recover some of the structures for genes involved in Mitochondria function.

In the real data analysis, we could not recover the exact causal relationships between variables that are shown in the KEGG database. This is due to the complexity of the biological structure and limited data samples. Moreover, the data may violate the normality assumption. Other robust methods such as the nonparanormal model (Liu *et al.* 2009, 2012) which implement a nonparametric transformation of the data may be applied to our framework. In addition, we expect more reliable structures can be obtained if we incorporate prior known information into our analysis by pre-setting some edges. We expect that more accurate and stable estimation can be achieved via incorporating both statistical and biological knowledge into the model development process. These will be incorporated into our future investigation. R code for implementing the method can be downloaded at http://www.stt.msu.edu/ ∼cui.

Acknowledgements

The authors thank Dr Ping-Shou Zhong for discussions on the proof of the theorem. The authors also thank the three anonymous reviewers for their insightful comments that greatly improved the manuscript.

Funding

This work was supported in part by grants from National Science Foundation (DMS-1209112 and IOS-1237969) and by a grant from National Natural Science Foundation of China (31371336).

Conflict of Interest: none declared.

References

Ali,R.A. *et al*. (2009) Markov equivalence for ancestral graphs. *Ann. Stat.*, **37**, 2808–2837.

Andersson,S.A. *et al*. (1997) A characterization of Markov equivalence classes for acyclic digraphs. *Ann. Stat.*, **25**, 505–541.

Cai,T. *et al*. (2013) Covariate-adjusted precision matrix estimation with an application in genetical genomics. *Biometrika*, **100**, 407–499.

Cheung,V.G. and Spielman,R.S. (2002) The genetics of variation in gene expression. *Nat. Genet.*, **32**, 522–525.

Chichering,D. (2002) Learning equivalence classes of Bayesian-network structures. *J. Mach. Learn. Res.*, **2**, 445–498.

Friedman,J. *et al*. (2008) Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**, 432–441.

Frydenberg,M. (1990) The chain graph Markov property. *Scand. J. Stat.*, **17**, 333–353.

Kalisch,M. and Bühlmann,P. (2007) Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *J. Mach. Learn. Res.*, **8**, 613–636.

Kalisch,M. *et al*. (2012) Causal inference using graphical models with the R package pcalg. *J. Stat. Soft.*, **47**, 1–26.

Knight,K. and Fu,W. (2000), Asymptotics for Lasso-type estimators. *Ann. Stat.*, **28**, 1356–1378.

Lee,W. and Liu,Y. (2012) Simultaneous multiple response regression and inverse covariance matrix estimation via penalized Gaussian maximum likelihood. *J. Multivar. Anal.*, **111**, 241–255.

Liu,H. *et al*. (2009), The nonparanormal: semiparametric estimation of high dimensional undirected graph. *J. Mach. Learn. Res.*, **10**, 2295–2328.

Liu,H. *et al*. (2012), High dimensional semiparametric Gaussian copula graphical models. *Ann. Stat.*, **40**, 2293–2326.

Meinshausen,N. and Bühlmann,P. (2006) High-dimensional graphs and variable selection with the lasso. *Ann. Stat.*, **34**, 1436–1462.

Pearl,J. (2009) *Causality*, 2nd edn, Cambridge University Press.

Richardson,T. and Spirtes,P. (2002) Ancestral graph Markov models. *Ann. Stat.*, **30**, 962–1030.

Schadt,E.E., *et al*. (2003) Genetics of gene expression surveyed in maize, mouse and man. *Nature*, **422**, 297–302.

Spirtes,P. *et al*. (2000) *Causation, Prediction, and Search*, 2nd edn, The MIT Press, One Rogers Street, Cambridge MA 02142–1209.

Verma,T. and Pearl,J. (1990) Equivalence and synthesis of causal models. In: Henrion,M. *et al* (eds.) *Uncertainty in Artificial Intellegence: Proceedings of the Sixth Conference*. Morgan Kaufman, San Francisco, CA, pp. 220–227.

Verma,T. and Pearl,J. (1992) An algorithm for deciding if a set of observed independencies has a causal explanation. In: Dubois,D. *et al* (eds.) *Uncertainty in Artificial Intelligence: Proceedings of the Eighth Conference*. Morgan Kaufman, San Francisco, CA, pp. 323–330.

Webster,J.A. *et al*. (2009), Genetic control of human brain transcript expression in Alzheimer disease. *Am. J. Hum. Genet.*, **84**, 445–458.

Yin,J. and Li,H. (2011) A sparse conditional Gaussian graphical model for analysis of genetical genomics data. *Ann. Appl. Stat.*, **5**, 2630–2650.

Yin,J. and Li,H. (2013) Adjusting for high-dimensional covariates in sparse precision matrix in estimation by l_1-penalizatio. *J. Multivar. Anal.*, **116**, 365–381.

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