Hindawi Publishing Corporation
BioMed Research International
Volume 2014, Article ID 629697, 9 pages
http://dx.doi.org/10.1155/2014/629697

Research Article

Inference of SNP-Gene Regulatory Networks by Integrating Gene Expressions and Genetic Perturbations

Dong-Chul Kim,[1] Jiao Wang,[2] Chunyu Liu,[3] and Jean Gao[1]

[1] *Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019, USA*
[2] *Beijing Genomics Institution at Wuhan, Wuhan 430075, China*
[3] *Department of Psychiatry, University of Illinois at Chicago, Chicago, IL 66012, USA*

Correspondence should be addressed to Jean Gao; gao@uta.edu

Received 28 January 2014; Accepted 9 May 2014; Published 9 June 2014

Academic Editor: Xing-Ming Zhao

In order to elucidate the overall relationships between gene expressions and genetic perturbations, we propose a network inference method to infer gene regulatory network where single nucleotide polymorphism (SNP) is involved as a regulator of genes. In the most of the network inferences named as SNP-gene regulatory network (SGRN) inference, pairs of SNP-gene are given by separately performing expression quantitative trait loci (eQTL) mappings. In this paper, we propose a SGRN inference method without predefined eQTL information assuming a gene is regulated by a single SNP at most. To evaluate the performance, the proposed method was applied to random data generated from synthetic networks and parameters. There are three main contributions. First, the proposed method provides both the gene regulatory inference and the eQTL identification. Second, the experimental results demonstrated that integration of multiple methods can produce competitive performances. Lastly, the proposed method was also applied to psychiatric disorder data in order to explore how the method works with real data.

1. Introduction

In order to understand more accurate causal relationships between a complex disease and genetic variations, we need to consider how the genotypic perturbations affect expression phenotypes that are potentially associated with a target disease. In other words, it is more crucial to look at the overall mechanisms considering a series of three factors, which include genetic variations, altering gene regulations, and caused diseases rather than partial mappings between them. Therefore it is important to evaluate how genetic perturbations affect genes on regulatory networks that are associated with a target disease phenotype. In practice, when biological networks are inferred with high throughput data, we have to consider not only the relationships among genes but also how genetic factors such as single nucleotide polymorphism (SNP) and copy number variation (CNV) can affect genes in gene regulatory network (GRN). Over the last decade, research for mapping genotype to expression phenotype or disease phenotype such as expression quantitative trait loci (eQTL) study and genome wide association study have been actively performed [1]. However, we are now required to do a network-based analysis with genotype data and gene expression because it is more effective in discovering underlying biological process from genotype to phenotype. In doing so, the analysis of SNP-gene regulatory networks (SGRN) will provide more definite relationships of genotypic causes and phenotypic effects so that it will facilitate prognosis and drug designs for therapies.

In this paper we propose a SGRN inference method. In order to identify regulatory interactions among genes, quite a number of network inference methods have been developed by using gene expression data such as gene microarray. Those methods can be generally classified into different theoretical categories: Boolean networks [2, 3], mutual information [4, 5], Bayesian networks (BN) [6, 7], and regression [8, 9]. As each method has its own advantages and limitations under different assumptions and network models such as acyclic or cyclic network and directed or undirected network, there should be trade-offs in inferences given different target network structure and applications [10]. For example, the MI-based approach is very simple and fast so that it can build a large scale network (e.g., genome wide scale) but it cannot estimate direction of edges. It produces worse performance than other approaches in detecting linear cascading structures [10]. The BN-based inference is limited to imply only acyclic network with high computational cost while the regression-based approach supports both directed and cyclic network, which are assumed in SGRN. In addition to directed network model, it should be considered that SGRN is different from conventional GRN inference. In SGRN inference, a gene can be regulated by SNPs as well as other genes, but SNPs are assumed to not be regulated by other SNPs. That is, a SNP cannot be a child node in the network.

Recently, a number of approaches have been suggested to infer SGRNs integrating genetic variation and gene expression data. Kim et al. [11] considered genetic perturbations, gene expression, and disease phenotypes together to find the causal genes to a disease. The electric circuit approach and heuristic search were used to infer SGRN where causal genes are mapped to SNP in the preliminary step before network inference. Keurentjes et al. [12] built a SNP-gene network associated with a particular phenotype, but this method also performed eQTL mapping (SNP-gene) to define the candidate regulator genes before genetic network construction. In addition, Kim and Xing [13] used lasso regression considering the case that a SNP is weakly associated with highly correlated multiple traits rather than a single trait. Chen et al. [14] focused on identifying which pathway among those already known pathways was more likely to be affected by changes of genotype and gene expression rather than inferring a new pathway. The related works we especially noted are the methods that are based on structural equation modeling (SEM) [15–18]. SEM allows us to not only incorporate eQTL information to gene expression in a single model but also identify eQTL simultaneously. However, Logsdon and Mezey [17] assumed that every gene has at least one eQTL, and eQTL mapping was performed by preprocessing but not in a network inference step. Cai et al. [18] introduced sparsity-aware maximum likelihood (SML), which can be potentially extended for eQTL identification. However, SNP-gene pairs were still given in evaluations and implementations of the SML algorithm.

In this paper, we proposed a novel method to infer SGRN where both eQTL identification and SGRN inference are performed simultaneously given a set of gene expression and genotype data without assuming eQTLs are known. The proposed method is based on SEM and multiple steps of edge filtering such as elastic net regression and iterative adaptive lasso. Basically SEM is a regression-based model which is likely to select as many variables causing an overfitting, so the sparsity is enforced by lasso (l_1-regularized least square estimation) considering the sparsity of biological network. Initial weights of edges are estimated by ridge regression [19] and elastic net regression [20], and then the second step is to identify final eQTLs from candidate SNPs selected in the first steps. In the last step, the final network is constructed by iterative adaptive lasso. The first two steps are to fix SNPs before selecting genes. In the third step, edges are selected by iteratively giving more penalties to the edge whose weight is relatively low until network structure is converged.

To evaluate the method, we explore the performance with a simulated data set, that is, generated from random networks with different number of samples and nodes and expected number of edges per node. The result shows that the method can achieve a high detection rate of true edges with low false discovery rate without eQTL information. In addition, to explore the performance in real expression phenotype and SNP data, the method was applied to the psychiatric disorder data. After genes and SNPs were selected from related Genome-Wide Association Study (GWAS), it was tested how the method identify true positive edges between genes and SNPs without eQTL information.

2. Method

2.1. Problem Definitions. We define the problem and notations here. Let $Y \in \mathbb{R}^{M_g \times N}$ denote the matrix of gene expression levels of M_g genes and N samples where a row vector $\mathbf{y}_i = \{y_{i1}, \ldots, y_{iN}\}$ is observed expression level of ith gene. X is $M_s \times N$ matrix to denote genotypes of individuals, where $x_{ij} \in \{1, 2, 3\}$ represents the number of minor alleles of ith SNP of jth sample as an element of matrix X supposing that the number of minor alleles should be zero, one, or two in real data. So, x_{ij} represents a relative quantity of minor alleles of samples. As a gene can be regulated by other genes and genetic variations (SNPs), we define SEM as

$$\mathbf{y}_i = \mathbf{b}_i Y + \mathbf{f}_i X + \mu_i + \varepsilon_i, \tag{1}$$

where $\mathbf{b}_i$ denotes ith row vector of square matrix $B \in \mathbb{R}^{M_g \times M_g}$; $\mathbf{f}_i$ denotes ith row vector of square matrix $F \in \mathbb{R}^{M_g \times M_s}$; μ_i is a model bias; and ε_i is a residual modeled as zero-mean Gaussian with a variance σ^2. As we assume there is no self-regulation (self-loop edge), $b_{ii} = 0, \forall i = 1, \ldots, M_g$, where b_{ii} denotes ith element of $\mathbf{b}_i$. The parameters of $\mathbf{b}_i$ and $\mathbf{f}_i$ decide the network structure defining the weight of regulation from every possible gene and SNP to a target gene i. For example, if there is no regulation relationship (directed edge) from jth gene to ith gene, b_{ij} is set to zero. Similarly f_{ij} has nonzero value as a weight of regulation from jth SNP to ith gene if jth SNP is identified as an eQTL for ith gene. It is assumed that each gene has at least one eQTL but it is unknown which SNP among a given set of SNPs is an eQTL for a target gene. Our goal in this model is to find B and F that best fit to observed gene expression and genotype data. To make the problem simpler, we remove μ_i from (1) by applying mean centering for row vectors $\mathbf{y}_i$ and $\mathbf{x}_i$ to have zero mean. The goal is to find $\mathbf{b}_i$ and $\mathbf{f}_i$ that minimize a residual ε_i, so (1) can be expressed in a least square minimization problem as

$$\arg\min_{\mathbf{b}_i, \mathbf{f}_i} \left\| \mathbf{y}_i - \mathbf{b}_i Y - \mathbf{f}_i X \right\|_2^2. \tag{2}$$

However, regression tends to select as many genes and SNPs as possible to explain the expression level of target gene i. To avoid the overfitting, sparse regression methods such as ridge regression, elastic net, and lasso are used.

2.2. The Algorithm. The method we propose is based on l_1-regularized linear regression known as lasso [21] that yields

```
procedure Elastic($Y, X, \widehat{\lambda}_1, \widehat{\lambda}_2, i, \varepsilon$) ▷ $\widehat{\lambda}_1$ and $\widehat{\lambda}_2$ are optimal parameters estimated by cross validation
    while err > $\varepsilon$ do
        $\mathbf{b}_i^{\text{old}} = \mathbf{b}_i, \mathbf{f}_i^{\text{old}} = \mathbf{f}_i$
        for $j \leftarrow 1, M_s$ do
            Update $f_{ij}$ via (12)
        end for
        Update $\mathbf{b}_i$ via (5)
        err = $\|\mathbf{b}_i^{\text{old}} - \mathbf{b}_i\|_2 + \|\mathbf{f}_i^{\text{old}} - \mathbf{f}_i\|_2$
    end while
    return $\mathbf{b}_i$ and $\mathbf{f}_i$
end procedure
```

Algorithm 1: Optimization for *elastic net* in Step 1-2.

a sparsity of variable selection. The algorithm consists of 3 steps, (i) *elastic net*, (ii) *lasso*, and (iii) *iterative adaptive lasso*. The first two steps are to decide F where SNPs are selected but their coefficients can be changed in the third step. Then, B is finalized by iterative adaptive lasso in the last step.

2.2.1. Ridge Regression (Step 1-1). In ridge regression, the coefficient values of irrelevant SNPs and genes to a target gene shrink to zero (but not exactly zero) while those of eQTLs and regulator genes of a target gene tend to be higher. Ridge regression of (2) is defined as

$$\underset{\mathbf{b}_i,\mathbf{f}_i}{\arg\min}\left\|\mathbf{y}_i - \mathbf{b}_iY - \mathbf{f}_iX\right\|_2^2 + \lambda_1\left\|\mathbf{b}_i\right\|_2^2 + \lambda_2\left\|\mathbf{f}_i\right\|_2^2. \tag{3}$$

Given penalty weights, λ_1 and λ_2, the optimal $\mathbf{b}_i$ and $\mathbf{f}_i$ can be obtained by closed form solution given by

$$\mathbf{f}_i = \left(\mathbf{y}_i - \mathbf{b}_iY\right)X^T\left(XX^T + \lambda_2 I\right)^{-1}, \tag{4}$$

$$\mathbf{b}_i = \left(\mathbf{y}_i - \mathbf{f}_iX\right)Y^T\left(YY^T + \lambda_1 I\right)^{-1}. \tag{5}$$

Replacing (5) for $\mathbf{b}_i$ in (4) yields

$$\mathbf{f}_i = \mathbf{y}_iS_1\left(XS_1 + \lambda_2 I\right)^{-1}, \tag{6}$$

where

$$S_1 = X^T - Y^T\left(YY^T + \lambda_1 I\right)^{-1}YX^T. \tag{7}$$

After calculating $\mathbf{f}_i$ first in (6) then (5) can be solved. In this manner, matrices B and F are estimated by computing each $\mathbf{b}_i$ and $\mathbf{f}_i$, $i = 1, \dots, M_g$. Parameters λ_1 and λ_2 that decide the degree of sparsity of B and F are determined by K-fold cross-validation. K is set to 5 in our experiments.

2.2.2. Elastic Net (Step 1-2). Note that zero weighted coefficient cannot be recovered back to nonzero in adaptive lasso of Step 3. Therefore, in order to carefully keep only SNPs that are more likely to be true eQTLs in $\mathbf{f}_i$, we give l_1-norm penalty to only $\mathbf{f}_i$ but not $\mathbf{b}_i$ using elastic net defined as

$$\underset{\mathbf{b}_i,\mathbf{f}_i}{\arg\min}\left\|\mathbf{y}_i - \mathbf{b}_iY - \mathbf{f}_iX\right\|_2^2 + \lambda_1\left\|\mathbf{b}_i\right\|_2^2 + \lambda_2\left\|\mathbf{f}_i\right\|_1. \tag{8}$$

As the objective function is convex, which guarantees a convergence, f_{ij} can be optimized by using coordinate descent iteration given parameters, λ_1 and λ_2. To find the optimal $\mathbf{f}_i$, the derivative of (8) with respect to f_{ij} is considered as follows:

$$\mathbf{f}_iXX_j^T - \mathbf{y}_iX_j^T + \mathbf{b}_iYX_j^T + \lambda_2\partial_{f_{ij}}\left\|\mathbf{f}_i\right\|_1. \tag{9}$$

Since the derivative of (8) with respect to $\mathbf{b}_i$ is the same as (5), $\mathbf{b}_i$ in (9) is substituted with (5), and then (9) is simplified to

$$\left(\mathbf{f}_{i(-j)}X_{(-j)} - \mathbf{y}_i\right)S_2 + f_{ij}\mathbf{x}_jS_2 - \lambda_2\partial_{f_{ij}}\left\|\mathbf{f}_i\right\|_1, \tag{10}$$

where

$$S_2 = \left(Y^T\left(YY^T + \lambda_1 I\right)^{-1}Y - I\right)\mathbf{x}_j^T; \tag{11}$$

$\mathbf{f}_{i(-j)}$ indicates row vector $\mathbf{f}_i$ whose jth element is removed, $X_{(-j)}$ denotes matrix X whose jth row is removed, and $\mathbf{x}_j$ is jth row vector of X. After defining $C_j = (\mathbf{f}_{i(-j)}X_{(-j)} - \mathbf{y}_i)S_2$ and $a_j = \mathbf{x}_jS_2$ in (10), the update rule in the coordinate descent algorithm is written as

$$f_{ij} = \begin{cases} \dfrac{\left(-C_j - \lambda_2\right)}{a_j} & \text{if } C_j < -\lambda_2, \\ 0 & \text{if } \left|C_j\right| \le \lambda_2, \\ \dfrac{\left(-C_j + \lambda_2\right)}{a_j} & \text{if } C_j > \lambda_2. \end{cases} \tag{12}$$

Algorithm 1 describes the procedures to solve (8) in Step 2. If f_{ij} is nonzero, jth SNP is a candidate eQTL for ith gene.

2.2.3. Lasso (Step 2). In order to finalize a SNP (a single nonzero f_{ij} of $\mathbf{f}_i$) for each gene i, we apply lasso to combined matrix of Y and X as follows:

$$\left\|\mathbf{y}_i - \mathbf{h}_iZ\right\|_2^2 + \lambda\left\|\mathbf{h}_i\right\|_1, \tag{13}$$

where

$$Z^T = \left[Y^T_{(-i)}, X^T_{(-k_i^*)}\right]. \tag{14}$$

k_i^* denotes indices of low vectors where $f_{ij} = 0$, $j \in k_i^*$. So, $X_{(-k_i^*)}$ is a matrix X whose k_i^* rows are removed. If the number of rows of $X_{(-k_i^*)}$ is greater than predefined heuristic number N_k (i.e., 5 in our experiments), only top N_k highest f_{ij} of absolute values of $\mathbf{f}_i$ but not all nonzero f_{ij} are selected for $X_{(-k_i^*)}$. In Step 2, we iteratively estimate $\mathbf{h}_i$, decreasing λ from a high value that lets $\mathbf{h}_i$ have a zero vector. Regardless of elements of $\mathbf{h}_i$ for $Y_{(-i)}$, we note only which element of $\mathbf{h}_i$ for $X_{(-k_i^*)}$ has a nonzero value first assuming that the corresponding candidate SNP to h_{ij} is more likely to regulate a target gene i if h_{ij} for a row vector of $X_{(-k_i^*)}$ has nonzero value earlier than other elements of $\mathbf{h}_i$ during λ decreases.

2.2.4. Adaptive Lasso (Subroutine of Step 3). Adaptive lasso is defined as

$$\underset{\mathbf{b}_i,\mathbf{f}_i}{\arg\min}\left\|\mathbf{y}_i - \mathbf{b}_iY - \mathbf{f}_iX\right\|_2^2 + \lambda_1\left\|\mathbf{b}_i\right\|_{1,\mathbf{w}_i^b} + \lambda_2\left\|\mathbf{f}_i\right\|_{1,\mathbf{w}_i^f}, \quad (15)$$

where

$$\left\|\mathbf{b}_i\right\|_{1,\mathbf{w}_i^b} = \sum_j^N\left|b_{ij}\cdot w_{ij}^b\right|, \qquad \left\|\mathbf{f}_i\right\|_{1,\mathbf{w}_i^f} = \sum_j^N\left|f_{ij}\cdot w_{ij}^f\right|. \quad (16)$$

In (16), penalty weights, vectors $\mathbf{w}_i^b$ and $\mathbf{w}_i^f$, are defined as

$$w_{ij}^b = \left|\widehat{b}_{ij}\right|^{-\alpha}, \quad w_{ij}^f = \left|\widehat{f}_{ij}\right|^{-\beta}, \quad \forall j = \left\{1,\ldots,M_g\right\}, \quad (17)$$

where $\widehat{b}_{ij}$ and $\widehat{f}_{ij}$ are estimated in Step 2 that yields a sparsity to $\mathbf{f}_i$ but not $\mathbf{b}_i$. Zero coefficient of $\widehat{\mathbf{f}}_i$ in Step 2 is not considered as an eQTL for gene i. So, zero $\widehat{f}_{ij}$ yields zero w_{ij}^f in (17), and then if w_{ij}^f is zero, f_{ij} will never have nonzero value in adaptive lasso of Step 3 (16). The parameters α and β decide how much previous estimation such as $\widehat{b}_{ij}$ or $\widehat{f}_{ij}$ is reflected to next estimation of b_{ij} or f_{ij}. Therefore, f_{ij} that has smaller penalty weight w_{ij}^f is more likely to have nonzero value. In addition, we consider a special case that α and β are set to zero supposing that (i) we do not give a penalty weight to b_{ij} or f_{ij} by setting w_{ij}^b or w_{ij}^f to 1 if $\widehat{b}_{ij}$ or $\widehat{f}_{ij}$ is nonzero and (ii) we do not estimate elements of $\mathbf{b}_i$ or $\mathbf{f}_i$ by setting w_{ij}^b or w_{ij}^f to infinity if $\widehat{b}_{ij}$ or $\widehat{f}_{ij}$ is zero. The solution is similar to Step 2 in which either $\mathbf{b}_i$ or $\mathbf{f}_i$ is optimized by coordinate descent algorithm but it is applied to solve both $\mathbf{b}_i$ and $\mathbf{f}_i$ in Step 3. Derivative of (15) with respect to b_{ij} yields

$$\begin{aligned}&\mathbf{b}_iY\mathbf{y}_j^T - \mathbf{y}_i\mathbf{y}_j^T + \mathbf{f}_iX\mathbf{y}_j^T + \lambda_1\partial_{\mathbf{b}_{ij}}\left\|\mathbf{b}_i\right\|_{1,\mathbf{w}_i^b}\\ &\quad = b_{ij}\mathbf{y}_j\mathbf{y}_j^T + \left(\mathbf{b}_{i(-j)}Y_{(-j)} - \mathbf{y}_i + \mathbf{f}_iX\right)\mathbf{y}_j^T + \lambda_1\partial_{b_{ij}}\left\|\mathbf{b}_i\right\|_{1,\mathbf{w}_i^b},\end{aligned} \quad (18)$$

where $\mathbf{b}_{i(-j)}$ indicates row vector $\mathbf{b}_i$ whose jth element is removed and $Y_{(-j)}$ denotes matrix Y whose jth row is removed. After setting $C_j^b = (\mathbf{b}_{i(-j)}Y_{(-j)} - \mathbf{y}_i + \mathbf{f}_iX)\mathbf{y}_j^T$ and $a_j^b = \mathbf{y}_j\mathbf{y}_j^T$, the update rule for b_{ij} is as follows:

$$b_{ij} = \begin{cases}\dfrac{\left(-C_j^b - w_{ij}^b\cdot\lambda_1\right)}{a_j^b} & \text{if } C_j^b < -w_{ij}^b\cdot\lambda_1,\\ 0 & \text{if } \left|C_j^b\right| \le w_{ij}^b\cdot\lambda_1,\\ \dfrac{\left(-C_j^b + w_{ij}^b\cdot\lambda_1\right)}{a_j^b} & \text{if } C_j^b > w_{ij}^b\cdot\lambda_1.\end{cases} \quad (19)$$

We can also estimate f_{ij} in similar way. After defining $C_j^f = (\mathbf{f}_{i(-j)}X_{(-j)} - \mathbf{y}_i + \mathbf{b}_iY)\mathbf{x}_j^T$ and $a_j^f = \mathbf{x}_j\mathbf{x}_j^T$, the update rule for f_{ij} is given as

$$f_{ij} = \begin{cases}\dfrac{\left(-C_j^f - w_{ij}^f\cdot\lambda_2\right)}{a_j^f} & \text{if } C_j^f < -w_{ij}^f\cdot\lambda_2,\\ 0 & \text{if } \left|C_j^f\right| \le w_{ij}^f\cdot\lambda_2,\\ \dfrac{\left(-C_j^f + w_{ij}^f\cdot\lambda_2\right)}{a_j^f} & \text{if } C_j^f > w_{ij}^f\cdot\lambda_2.\end{cases} \quad (20)$$

When $\mathbf{b}_i$ and $\mathbf{f}_i$ are updated, updated single element b_{ij} or f_{ij} immediately affects updating the next elements. In addition, updating order of elements can be changed since convex objective function is converged in any order of elements to update. Algorithm 2 shows the optimization procedure of adaptive lasso.

2.2.5. Iterative Adaptive Lasso (Step 3). Even if $\mathbf{b}_i$ and $\mathbf{f}_i$ are estimated in Steps 1 and 2, there should be still many false positive edges yet. The primary goal of Steps 1 and 2 is to carefully get rid of only edges that are more unlikely to be true positive edges. So, instead of simply applying adaptive lasso, we developed iterative adaptive lasso to improve the performance of naive adaptive lasso. The motivation of iterative adaptive lasso is that the coefficient value of the variable considerably depends on the value of α and β which are fixed to 1 and 0.5 in [17, 18], respectively. In iterative adaptive lasso, adaptive lasso is iteratively applied incrementally changing α and β until there is no more change in the total number of selected edges of B and F so that more coefficients of irrelevant variables can be shrunk to zero.

Algorithm 3 presents a detailed procedure of iterative adaptive lasso. $\widehat{B}$ and $\widehat{F}$ estimated in Step 2 are used as arguments. On line 2, B and F are initialized by ridge regression. Λ_1^R is a vector of optimal λ_1 for B^R estimated by Ridge regression but there is no penalty to F^R (i.e. $\Lambda_2^R = 0$). When F^R is estimated, only non-zero elements of $\widehat{F}$ that is estimated in Step 2 are updated. On line 5, B and F are estimated by adaptive lasso in order that elements of B are updated by weights of B^R (i.e. b_{ij} that has a small value can shrink to zero). Before line 7 starts, $\widehat{\Lambda}_1$ (a vector of $\widehat{\lambda}_1$ for B on line 10) is estimated by cross validation of adaptive lasso. On line 7–14, more elements of B shrink to zero increasing α.

```
procedure ADAPTIVE LASSO($Y, X, \widehat{\lambda}_1, \widehat{\lambda}_2, i, \alpha, \beta, \widehat{\mathbf{b}}_i, \widehat{\mathbf{f}}_i$) ▷ $\widehat{\lambda}_1$ and $\widehat{\lambda}_2$ are optimal parameters preliminary estimated by cross validation
    Compute $\mathbf{w}_i^b$ and $\mathbf{w}_i^f$ ($w_{ij}^b = (\widehat{b}_{ij})^{-\alpha}, w_{ij}^f = (\widehat{f}_{ij})^{-\beta}$)
    while err > $\varepsilon$ do
        $\mathbf{b}_i^{\text{old}} = \mathbf{b}_i, \mathbf{f}_i^{\text{old}} = \mathbf{f}_i$
        for $j \leftarrow 1, M_g$ do
            Update $b_{ij}$ via (19)
        end for
        for $j \leftarrow 1, M_s$ do
            Update $f_{ij}$ via (20)
        end for
        err = $\left\|\mathbf{b}_i^{\text{old}} - \mathbf{b}_i\right\|_2 + \left\|\mathbf{f}_i^{\text{old}} - \mathbf{f}_i\right\|_2$
    end while
    return $\mathbf{b}_i$ and $\mathbf{f}_i$
end procedure
```

ALGORITHM 2: Optimization for adaptive lasso as a subroutine of Step 3.

```
procedure ITERATIVE ADAPTIVE LASSO($Y, X, \widehat{B}, \widehat{F}$) ▷ Ne($B$) denote the number of non-zero elements in $B$ and $F$
    $[B^R, F^R] = Ridge(Y, X, \widehat{\Lambda}_1^R, \widehat{F})$
    $\alpha = 1, \beta = 1$
    for $i \leftarrow 1, M_g$ do
        $[\mathbf{b}_i, \mathbf{f}_i] = AdaptiveLasso(Y, X, \lambda_1 = 0.001, \lambda_2 = 0, i, \alpha, \beta, \mathbf{b}_i^R, \mathbf{f}_i^R)$
    end for
    while Ne($B$) are decreased by increased $\alpha$ do
        while Ne($B$) are decreased do
            for $i \leftarrow 1, M_g$ do
                $[\mathbf{b}_i, \mathbf{f}_i] = AdaptiveLasso(Y, X, \widehat{\lambda}_1, \lambda_2 = 0, i, \alpha, \beta, \mathbf{b}_i, \mathbf{f}_i)$
            end for
        end while
        $\alpha = \alpha + 1$
    end while
    return $B$ and $F$
end procedure
```

ALGORITHM 3: Iterative adaptive lasso in Step 3.

The second *while* loop updates B until no change in $N_e(B)$. Once the second *while* loop is terminated, α is increased, and then the second loop is performed again. If the second *while* loop is terminated without any change of $N_e(B)$, the first *while* loop is terminated.

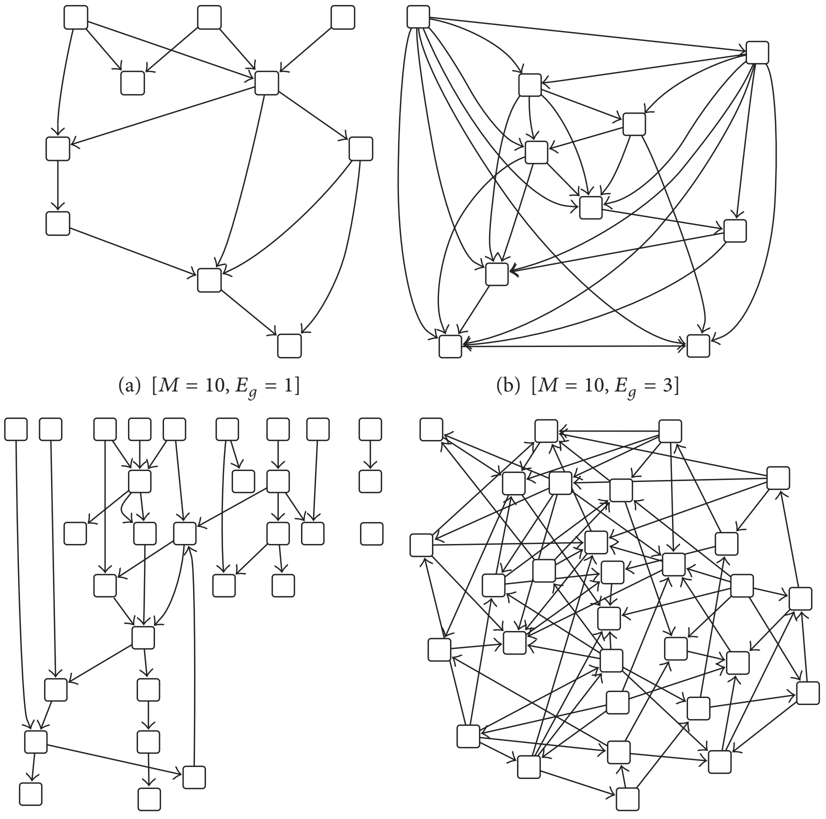
3. Results

3.1. Simulation Studies. To evaluate the proposed method, we first perform simulations based on randomly generated acyclic networks. The simulation settings are similar to those in [17, 18]. M denotes the number of genes and SNPs and is set to 10, 20, and 30. $M \times N$ matrix B is initialized to zero matrix where N is a sample size; then elements of B are randomly selected as directed edges. The selected b_{ij} has random coefficient value uniformly distributed over 0.5~1 or −0.5~ −1. Since we consider a single eQTL per gene ($E_s = 1$), a single element (f_{ii}) is selected from each row vector ($\mathbf{f}_i$). So, F is a diagonal matrix. x_{ij} is randomly set as 1, 2, or 3 with the probabilities 0.25, 0.5, and 0.25, respectively. Y is generated by calculating $Y = (I - B)^{-1}(FX + E)$, where E_{ij} is generated from Gaussian distribution with zero mean and variance 0.01. The number of samples for each network size is $N = 100, 200, 300, 400,$ and 500. The number of edges per gene on average is set to $E_g = 1, 2,$ and 3. Given data Y and X, performances of predicting B and F are evaluated by comparing true network and inferred network.

Figure 1 displays the examples of networks, where SNP nodes are excluded. For the evaluation, true positive (TP), false positive (FP), true negative (TN), and false negative (FN) edges are counted to measure the accuracy criteria such as true positive rate (TPR) and false discovery rate (FDR) that are defined as

(i) TPR = TP/(TP + FN),

(ii) FDR = FP/(TP + FP).



(a) $[M = 10, E_g = 1]$

(b) $[M = 10, E_g = 3]$

(c) $[M = 30, E_g = 1]$

(d) $[M = 30, E_g = 3]$

Figure 1: Example of simulated networks with different parameter settings. M and E_g indicate the number of genes and expected number of edges per node, respectively.

Table 1: TPR and FDR of SML, IAL1, and IAL2.

N	M	TPR			FDR		
		SML	IAL1	IAL2	SML	IAL1	IAL2
100	10	0.9888	1.0000	0.9742	0.0860	0	0.0104
	20	0.9980	1.0000	0.9448	0.0503	0	0.0292
	30	0.9951	1.0000	0.8936	0.0364	0	0.0754
500	10	0.9967	1.0000	1.0000	0.0704	0	0
	20	0.9850	1.0000	0.9436	0.0400	0	0.0369
	30	1.0000	1.0000	0.9128	0.0016	0	0.0562

Expected number of edges per node is $E_g = 2$ and 10 replicates of random network are used. N and M indicate the number of samples and genes, respectively.

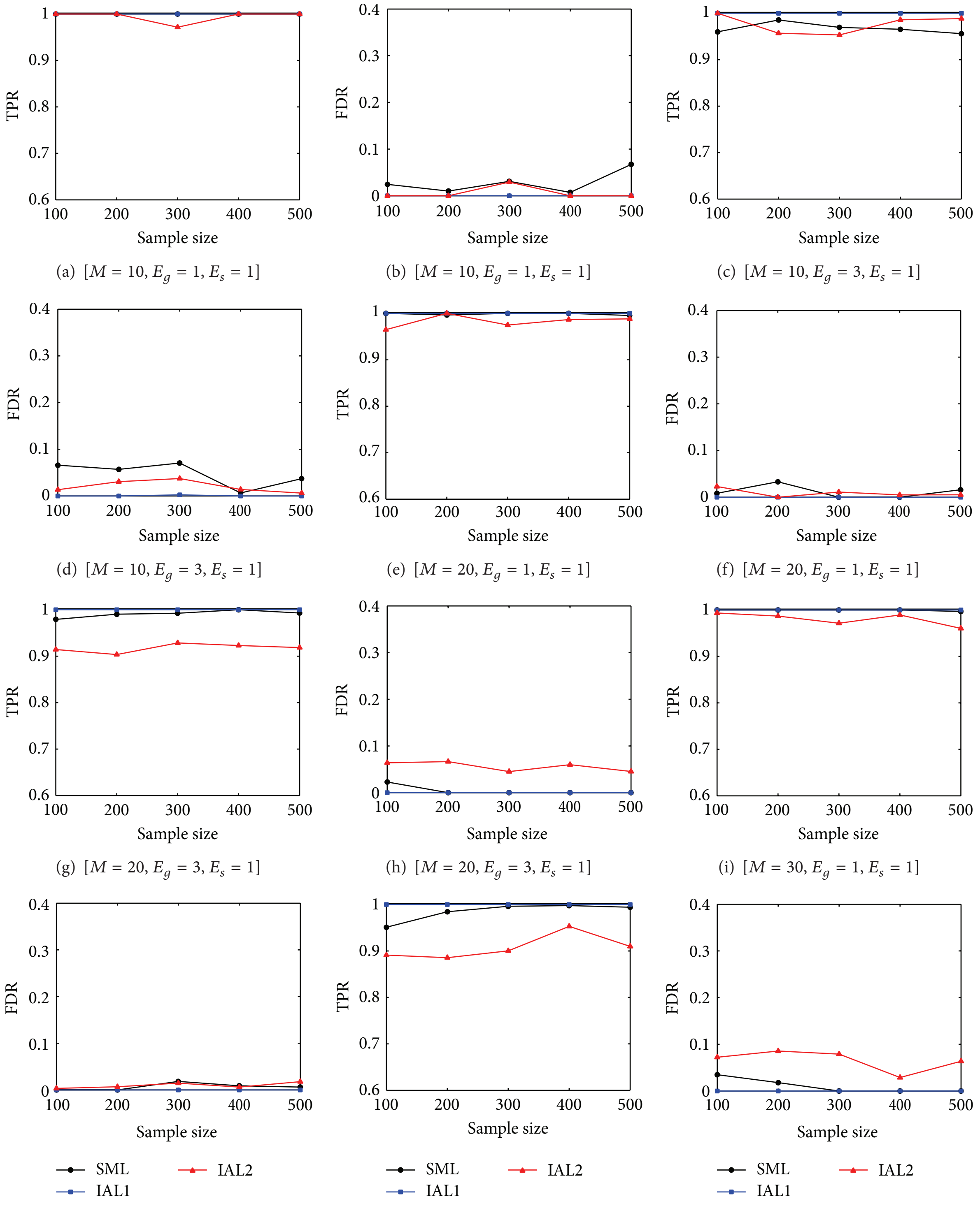
In order to evaluate our method, IAL is compared to SML [18]. As SML infers only B with known nonzero element indices of F, we consider two versions of IAL, IAL without eQTL information and IAL with eQTL information, where Steps 1 and 2 are skipped and only Step 3 is performed with nonzero element index of $\mathbf{f}_i$. SML is tested by using the code the author implemented in [18]. The abbreviations of algorithms to compare in Figure 2 and Table 1 are listed below:

(i) SML: sparsity-aware maximum likelihood algorithm with eQTL information [18],

(ii) IAL1: IAL with eQTL information,

(iii) IAL2: IAL without eQTL information.

Ten replicate simulations are performed and each simulation has a different topology. The results of the different settings (M and E_g) are displayed in Figure 2. It is shown that IAL1 is superior to SML in all data sets regardless of sample size. We also note that TPR of IAL2 is higher than 0.9 and FDR is less than 0.1 on average in any sample size. It validates that the proposed IAL works very effectively when eQTL is known. In addition, the performance of IAL1 is consistent in different sample sizes while the performance of SML tends to be decreased with small sample size and complicated network ($E_g = 3$). In network inference, it is known that the performance of inference is very sensitive to the network size and density. In the inference of densely connected and large networks, the computational cost will exponentially increase and the FDR may increase because there are more possible variables that may explain a target node better than true regulators. IAL1 performed consistently in all three different network sizes while the performance of SML is affected by the network size in dense networks ($E_g = 3$). However, IAL2 shows consistent TPRs and FDRs in all three different network sizes when the network density is normal ($E_g = 1$) while TPR of IAL2 in Figures 2(g) and 2(k) is lower than Figure 2(c) and also FDR increases in



(a) $[M = 10, E_g = 1, E_s = 1]$

(b) $[M = 10, E_g = 1, E_s = 1]$

(c) $[M = 10, E_g = 3, E_s = 1]$

(d) $[M = 10, E_g = 3, E_s = 1]$

(e) $[M = 20, E_g = 1, E_s = 1]$

(f) $[M = 20, E_g = 1, E_s = 1]$

(g) $[M = 20, E_g = 3, E_s = 1]$

(h) $[M = 20, E_g = 3, E_s = 1]$

(i) $[M = 30, E_g = 1, E_s = 1]$

(j) $[M = 30, E_g = 1, E_s = 1]$

(k) $[M = 30, E_g = 3, E_s = 1]$

(l) $[M = 30, E_g = 3, E_s = 1]$

Figure 2: True positive rate and false discovery rate under different numbers of edges and nodes.

Table 1 when the network size increases in more dense networks ($E_g = 2$).

The result shows that the performance is better in sparse networks ($E_g = 1$) than dense networks ($E_g = 3$) because a complicated structure is more likely to cause false positive edges because of indirect regulations. For example, TPRs in Figures 2(a), 2(e), and 2(i) are much better than in Figures 2(c), 2(g), and 2(k). Similarly FDR is quite increased with $E_g = 3$ in Figures 2(d), 2(h), and 2(l) compared to the case of $E_g = 1$ in Figures 2(b), 2(f), and 2(j).

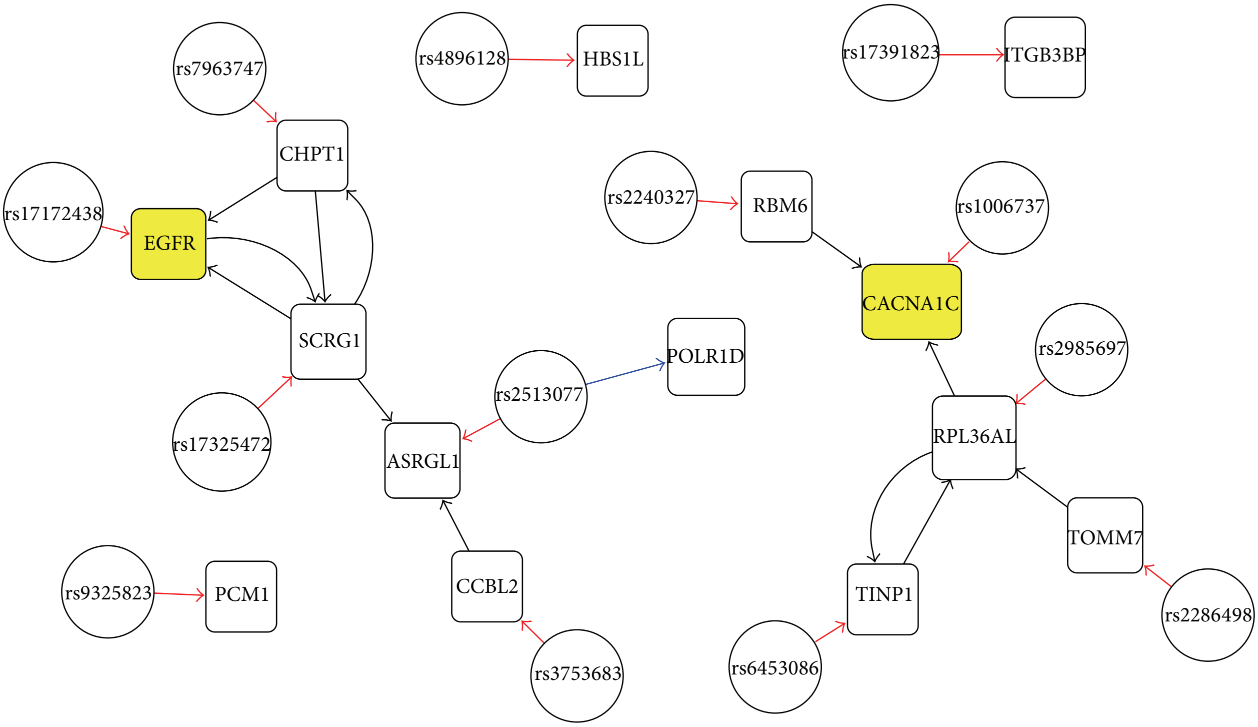


Figure 3: The inferred SGRN with 14 pairs of gene and SNP selected from [22–24].

Overall results imply that the proposed IAL1 works perfectly with known F in any network size and density. It means that the performance of IAL2 is significantly affected by false positive inference of F in steps 1 and 2 because of unknown F. More precisely $\mathbf{b}_i$ without sparsity in step 2 is more likely to have false positive nonzero elements even though a number of candidate elements of $\mathbf{b}_i$ are filtered in step 1. Therefore, the selection of nonzero element of $\mathbf{b}_i$ in IAL2 is the most critical part since IAL1 is able to correctly infer B only if F is given as eQTL information.

3.2. Experiments with Psychiatric Disorder Data. In this section, the proposed method is applied to real gene expression and genotype data for psychiatric disorder. In the application to real data, we explore the performance of GRN inferences and eQTL identifications through the inferred networks. As far as we know, the proposed method is the first solution to provide both GRN inference and eQTL identification. Thus, the performance comparison with other methods was not performed. The psychiatric disorder data consists of gene expression data of 25833 genes and 852963 SNPs for 131 samples, which were measured from human brain. Since we focus on the network inference but not gene selection, the network construction is performed with a predefined set of genes and SNPs that are selected by preliminary test of multiple sets of genes and eQTLs based on related GWAS for psychiatric disorders. The result of SGRN inference is displayed in Figure 3 where two yellow colored genes, EGFR and CACNA1C, are selected from [23, 24] and the rest of two pairs are from [22]. In applying IAL2 to the data, the weights of α and β are set to 0.5 instead of 1. Otherwise, $N_e(\mathbf{f}_i)$ tends to be zero. The reason for this is that gene variables are more correlated with their eQTLs because generally eQTLs are independently selected to other genes. In Figure 3, SNP and gene are distinguished by node shape, and a red edge indicates a correct edge from eQTL to corresponding gene. A blue edge represents false positive eQTL mapping. For eQTL identification, one false positive edge appears and thirteen true positive edges are detected (TPR = 0.9286, FDR = 0.0714).

4. Discussion

The most difficult part in network inference is to identify directions of edges. In the adjacency matrix B, both B_{ij} and B_{ji} could have a high coefficient value. In this case, regression-based methods tend to show better performance than MI-based methods because candidate edges are evaluated together in regression-based methods but each edge is independently evaluated to other edges in MI-based methods. Despite the advantage, the regression-based method needs to be integrated with other methods that can provide different information of structure. Another issue to improve in IAL is the computational cost to estimate two different λs per each row. Intuitively, a searched optimal λ per each row of B and F should provide a better result but it causes a high computation cost. Lastly, we also assumed that a gene has at least a single eQTL given a set of genes and SNPs, but multiple eQTLs should be considered and a gene may not have any eQTL in practice. Thus, the multiple eQTL of a gene is a future work in SGRN inference.

5. Conclusion

In this paper, we proposed a novel network inference method that provides both eQTL identification and network construction of both genes and SNPs. In order to understand gene regulatory mechanisms for a target disease phenotype, the regulatory network inference needs to consider effect of genetic variation and expression phenotype together but not only gene expression data. To achieve the high quality of reliable inference with better TPR and FDR, three different regression skills are integrated. Ridge regression and elastic net are used to remove more likely false positive edges and select eQTL as preliminary steps, and then the finial network is estimated by iterative adaptive lasso removing more false positive edges between genes. Through the experiments with synthetic data, it was demonstrated that IAL1 outperforms SML in SGRN inference and also IAL2 performs eQTL identification effectively. The method was also applied to psychiatric disorder data. Using the genes and eQTLs selected from GWAS of psychiatric disorder, we explored the ability of eQTL identification through inferred SGRN.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] A. C. Nica and E. T. Dermitzakis, "Expression quantitative trait loci: present and future," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 368, no. 1620, 2013.

[2] J. Liang and J. Han, "Stochastic Boolean networks: an efficient approach to modeling gene regulatory networks," *BMC Systems Biology*, vol. 6, article 113, Article ID 20120362, 2012.

[3] B. Vasić, V. Ravanmehr, and A. R. Krishnan, "An information theoretic approach to constructing robust boolean gene regulatory networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 1, pp. 52–65, 2012.

[4] R. de Matos Simoes and F. Emmert-Streib, "Bagging statistical network inference from large-scale gene expression data," *PLoS ONE*, vol. 7, no. 3, Article ID e33624, 2012.

[5] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring connectivity of genetic regulatory networks using information-theoretic criteria," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, no. 2, pp. 262–274, 2008.

[6] N. Xuan, M. Chetty, R. Coppel, and P. Wangikar, "Gene regulatory network modeling via global optimization of high-order dynamic Bayesian network," *BMC Bioinformatics*, vol. 13, no. 1, article 131, 2012.

[7] D.-C. Kim, X. Wang, C.-R. Yang, and J. Gao, "Learning biological network using mutual information and conditional independence," *BMC Bioinformatics*, vol. 11, supplement 3, article S9, 2010.

[8] G. Geeven, R. E. van Kesteren, A. B. Smit, and M. C. M. de Gunst, "Identification of context-specific gene regulatory networks with GEMULA-gene expression modeling using LAsso," *Bioinformatics*, vol. 28, no. 2, pp. 214–221, 2012.

[9] M. Gustafsson, M. Hörnquist, and A. Lombardi, "Constructing and analyzing a large-scale gene-to-gene regulatory network-lasso-constrained inference and biological validation," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 2, no. 3, pp. 254–261, 2005.

[10] D. Marbach, J. C. Costello, R. Küffner et al., "Wisdom of crowds for robust gene network inference," *Nature Methods*, vol. 9, no. 8, pp. 796–804, 2012.

[11] Y.-A. Kim, S. Wuchty, and T. M. Przytycka, "Identifying causal genes and dysregulated pathways in complex diseases," *PLoS Computational Biology*, vol. 7, no. 3, Article ID e1001095, 2011.

[12] J. J. B. Keurentjes, J. Fu, I. R. Terpstra et al., "Regulatory network construction in Arabidopsis by using genome-wide gene expression quantitative trait loci," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 5, pp. 1708–1713, 2007.

[13] S. Kim and E. P. Xing, "Statistical estimation of correlated genome associations to a quantitative trait network," *PLoS Genetics*, vol. 5, no. 8, Article ID e1000587, 2009.

[14] L. Chen, L. Zhang, Y. Zhao et al., "Prioritizing risk pathways: a novel association approach to searching for disease pathways fusing SNPs and pathways," *Bioinformatics*, vol. 25, no. 2, pp. 237–242, 2009.

[15] M. Xiong, J. Li, and X. Fang, "Identification of genetic networks," *Genetics*, vol. 166, no. 2, pp. 1037–1052, 2004.

[16] B. Liu, A. de la Fuente, and I. Hoeschele, "Gene network inference via structural equation modeling in genetical genomics experiments," *Genetics*, vol. 178, no. 3, pp. 1763–1776, 2008.

[17] B. A. Logsdon and J. Mezey, "Gene expression network reconstruction by convex feature selection when incorporating genetic perturbations," *PLoS Computational Biology*, vol. 6, no. 12, Article ID e1001014, 2010.

[18] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations," *PLoS Computational Biology*, vol. 9, no. 5, Article ID e1003068, 2013.

[19] A. E. Hoerl and R. W. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[20] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.

[21] R. Tibshirani, "Regression shrinkage and selection via the lasso: a retrospective," *Journal of the Royal Statistical Society B: Methodological*, vol. 58, no. 1, pp. 267–288, 1996.

[22] C. Liu, L. Cheng, J. A. Badner et al., "Whole-genome association mapping of gene expression in the human prefrontal cortex," *Molecular Psychiatry*, vol. 15, no. 8, pp. 779–784, 2010.

[23] P. Sklar, J. W. Smoller, J. Fan et al., "Whole-genome association study of bipolar disorder," *Molecular Psychiatry*, vol. 13, no. 6, pp. 558–569, 2008.

[24] M. A. R. Ferreira, M. C. O'Donovan, Y. A. Meng et al., "Collaborative genome-wide association analysis supports a role for ANK3 and CACNA1C in bipolar disorder," *Nature Genetics*, vol. 40, no. 9, pp. 1056–1058, 2008.