IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 4, 2020, accepted February 21, 2020, date of publication February 27, 2020, date of current version March 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2976743

A Sparse Bayesian Learning Method for Structural Equation Model-Based Gene Regulatory Network Inference

YAN LI[1,2], DAYOU LIU[1,2], JIANFENG CHU[1], YUNGANG ZHU[1,2], JIE LIU[1,2], AND XIAOCHUN CHENG[3]

[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China
[3]Department of Computer Communications, Middlesex University, London NW4 4BT, U.K.

Corresponding authors: Jie Liu (liu_jie@jlu.edu.cn) and Xiaochun Cheng (x.cheng@mdx.ac.uk)

This work was supported by the National Natural Science Foundation of China under Grant 61502198, Grant 61572226, Grant 61472161, and Grant 61876069.

ABSTRACT Gene regulatory networks (GRNs) are underlying networks identified by interactive relationships between genes. Reconstructing GRNs from massive genetic data is important for understanding gene functions and biological mechanism, and can provide effective service for medical treatment and genetic research. A series of artificial intelligence based methods have been proposed to infer GRNs from both gene expression data and genetic perturbations. The accuracy of such algorithms can be better than those models that just consider gene expression data. A structural equation model (SEM), which provides a systematic framework integrating both types of gene data conveniently, is a commonly used model for GRN inference. Considering the sparsity of GRNs, in this paper, we develop a novel sparse Bayesian inference algorithm based on Normal-Equation-Gamma (NEG) type hierarchical prior (BaNEG) to infer GRNs modeled with SEMs more accurately. First, we reparameterize an SEM as a linear type model by integrating the endogenous and exogenous variables; Then, a Bayesian adaptive lasso with a three-level NEG prior is applied to deduce the corresponding posterior mode and estimate the parameters. Simulations on synthetic data are run to compare the performance of BaNEG to some state-of-the-art algorithms, the results demonstrate that the proposed algorithm visibly outperforms the others. What's more, BaNEG is applied to infer underlying GRNs from a real data set composed of 47 yeast genes from Saccharomyces cerevisiae to discover potential relationships between genes.

INDEX TERMS Sparse Bayesian learning, high-dimensional data, gene regulatory network, gene expression data, structural equation model.

I. INTRODUCTION

A gene is a segment of DNA which is the basic physical and functional unit of heredity. Genes direct the synthesis of functional molecules such as proteins and functional RNA, and thereby determine biological functions and phenotypes. This regulation is mainly implemented via the process of gene expression, including transcription and translation [1]. With the development of high throughput technologies such as DNA microarray and RNA-seq, tons of comprehensive genomic data such as the genome-wide gene expression data and gene variations in biological individuals can be easily obtained via experimental methods [2]. A large amount of genomic data have been reported in prior literatures [3]–[5], and several public repositories (such as the Gene Expression Omnibus (GEO)) have been built to provide service for gathering genomic data from bacterias, yeasts, plants and humans.

Genes in living organisms usually interact with each other and act together rather than working in isolation. Gene expression reflects regulatory relationships among individual genes, and can be taken full advantages to form underlying GRNs [6]–[8]. Delineating the structure of a GRN is of significant importance for understanding gene functions, cell physiologies and biological mechanisms. Additionally,

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Aljawarneh.

in practical applications, while several intelligent approaches have been developed to implement computer-aided diagnosis [9] and treatment [10], and help constructing better healthcare systems [11]–[13], GRNs can provide theoretical basis for various medical services at the molecular biology level (such as genetic testing service, genetic diagnosis service and molecular targeted therapy).

Artificial intelligence is an important technology for big data analysis and has been applied successfully in many fields and scenarios [14]–[16]. Since the explosive amount of genetic data, machine learning is also essential to tractable GRN inference. Much progress has been made to infer GRN structures from gene expression data. In early stage, Boolean networks are popular for GRN inference, in which the state of a gene is represented by a Boolean variable and the interactions between genes are represented by Boolean functions determining the state of a gene on the basis of the states of some other regulatory genes [17]–[19]. Boolean models are simple but cannot reflect quantitative biochemical details. Information theoretic methods are used to infer GRNs by detecting the correlations or mutual information between genes [20]–[22]. This kind of methods have heavy computational burden for large data sets and cannot directly infer GRNs with feedback loops. As for approaches based on Gaussian graphical models, undirected GRNs can be determined by the precision matrix [23], [24], and Bayesian networks are employed to infer directed *acyclic* GRNs (DAGs) [25]–[27]. Differential equations [28], [29] and regression models [30] have been widely used to identify both DAGs and directed *cyclic* GRNs (DCGs) by estimating the adjacency matrices, and a series of related inference algorithms have been developed in past a few years.

Basu *et al.* [31] built a computationally efficient iterative random forest (iRF) algorithm to search for high-order interactions, which can also be used for inferring GRNs from Boolean gene expression data. The iRF algorithm adopted a bagging step to assess the stability of recovered interactions, which allows robust identification with respect to small bootstrap perturbations in the data. However, iRF didn't consider the genetic perturbations yet, and is not applicable for continuous gene expression data, the data discretization process may cause information loss. Several methods have been developed to infer GRNs by exploiting genetic perturbations, including the algorithms based on Bayesian networks [32], [33], the causal models based on likelihood test [34], [35] and the approaches based on SEMs. Among them, the approaches based on SEMs have attracted a lot of research attentions. Different from iRF, GRNs modeled with SEMs takes genetic perturbations into considerations explicitly. The gene expression data are treated as endogenous variables, and the genetic variations observed at eQTLs are generally viewed as genetic perturbations, which are treated as exogenous variables. Due to the character of SEMs, the regulatory effects of both types of variables on each gene can be analyzed simultaneously. Besides, SEMs can provide more accurate GRN prediction than iRF by supporting inference based on continuous gene expression data. Furthermore, the topological structures of GRNs are depicted by adjacency matrices, which makes it possible to directly infer both DAGs and DCGs from SEMs.

The dimension of gene expression data is usually high, it is difficult to process them without any constraints. As discussed in [36], [37], GRNs or more general biochemical networks are sparse, meaning that a gene directly regulates or is regulated by a small number of genes relative to the total number of genes in the network. Motivated by this fact, the network sparsity constraints need to be incorporated into the inference of GRNs modeled with SEMs. In 2004, Xiong *et al.* [38] proposed to model GRNs with SEMs, whereafter several related algorithms were put forward successively [39]–[41]. Cai *et al.* [40] proposed a systematic inference algorithms for sparse SEMs named SML to infer GRNs by exploiting both gene expression data and eQTL data, which was proved to significantly outperforms other previous algorithms [39], [41]. Subsequently, Dong *et al.* [42] formulated a linear regression model from an SEM and developed an iterative Bayesian inference algorithm named LRBI to infer GRNs. More recently, Chen and Ren [43] built large systems of SEMs by coming up with a 2SPLS algorithm. They obtained the consistent estimation via ridge regression at the first stage, and then employed adaptive lasso at the second stage to achieve the consistent variable selection. The simulation results in [42] elucidated that LRBI has better performance than SML in terms of power of detection (PD) whereas SML performs better than LRBI in terms of false discover rate (FDR). And in [43], the simulation results demonstrated that the 2SPLS algorithm has better PD than SML and lower FDR than SML when the sample size is relatively smaller, whereas for bigger sample sizes, the FDR of 2SPLS is worse than SML.

By reviewing the simulation studies in the above studies, we found that in general none of the above state-of-the-art algorithms developed for sparse SEMs has completely better performance than the others. Motivated by this, in this paper, we focus on the inference of GRNs modeled with SEMs and develop a novel algorithm named BaNEG to improve the performance of existing methods. We first reparameterize the SEM by merging the exogenous and endogenous variables, and then develop a Bayesian adaptive lasso inference algorithm with hierarchical NEG prior to solve the reparameterized linear type models. Several simulations are conducted to compare the performance of our proposed BaNEG algorithm with three state-of-the-art algorithms: LRBI [42], SML [40] and 2SPLS [43]. The results demonstrate that BaNEG has similar performance with LRBI [42] in terms of PD, which outperforms SML [40] and 2SPLS [43] visibly. In the meantime, the FDR of BaNEG generally outperforms all the other algorithms.

II. METHODS

A. GRNS MODELED WITH SEMS

As in [39], [40], both gene expression data and genetic perturbations(such as eQTLs) can be integrated into SEMs

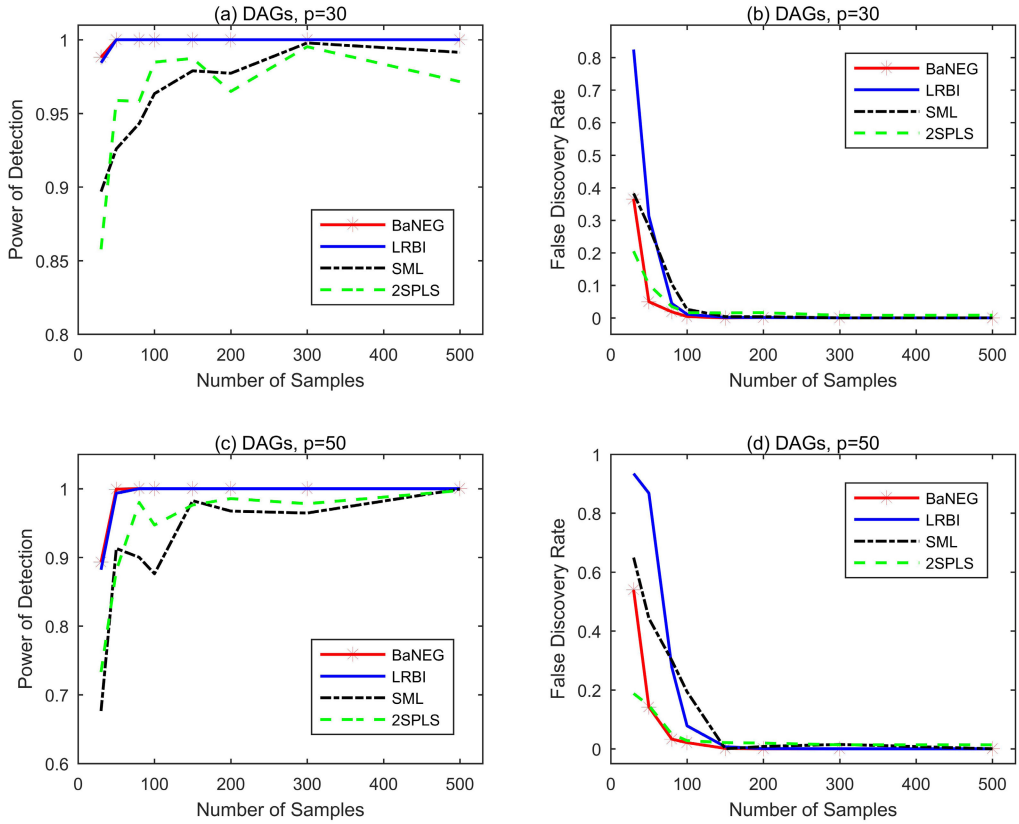

FIGURE 1. PD and FDR of BaNEG, LRBI, SML and 2SPLS for DAGs with 30 genes and 50 genes.

to model corresponding GRNs. We consider an SEM with p endogenous variables and q exogenous variables sampled from N individuals, the variables here represent gene expression levels of p genes and genotypes of q variant *cis*-eQTLs, respectively. We use *cis*-eQTLs here mainly because the empirical evidence proved that local genetic polymorphism tends to have larger effects than *trans*-eQTL [44], [45]. The expression levels of these p genes can be measured by technologies such as cDNA microarray and RNA-seq. Let $\mathbf{Y} := [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_p]$ be an $N \times p$ matrix, in which $\mathbf{y}_i := [y_{1i}, y_{2i}, \cdots, y_{Ni}]^T, i = 1, \cdots, p$ denote expression levels of the ith gene from N individuals. Let $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_q]$ be an $N \times q$ matrix, in which $\mathbf{x}_j := [x_{1j}, x_{2j}, \cdots, x_{Nj}]^T, j = 1, \cdots, q$ denote genotypes of the jth *cis*-eQTL from N individuals. Thus an SEM can be represented as a set of structural equations, which could be integrated as follows:

$$\mathbf{Y} = \mathbf{YB} + \mathbf{XF} + \mathbf{E}, \tag{1}$$

where the $p \times p$ matrix $\mathbf{B}$ denotes the topological structure of a GRN inferred from N observations. In $\mathbf{B}$, the value of each entry B_{ij} represents regulatory effect of the ith gene on the jth gene. It is often assumed that a gene has no effect on itself, which implies $B_{ii} = 0$ for $i = 1, \cdots, p$. The $q \times p$ matrix $\mathbf{F}$ denotes causal effects of the *cis*-eQTLs, in which F_{km} represents effect of the kth *cis*-eQTL on the mth gene. For the uniquely identifiability of the SEMs, as stated in [40], [42], [43], we assume that each gene in the GRN has a unique nonempty set of *cis*-eQTLs, so q is larger than or equal to p. $\mathbf{E}$ is an $N \times p$ matrix capturing the residual error terms. It is assumed that $\mathbf{X}$ and $\mathbf{E}$ are independent with each other.

For convenient calculation, model (1) could be split into p structural equations, each represents regulatory effects of all endogenous variables and exogenous variables on one gene. The ith model can be expressed as follows,

$$\mathbf{y}_i = \mathbf{Y}_{-i}\mathbf{b}_i + \mathbf{X}\mathbf{f}_i + \mathbf{e}_i, \quad i = 1, \cdots, p, \tag{2}$$

where $N \times 1$ vector $\mathbf{y}_i$ is expression levels of the ith gene measured from N individuals. $N \times (p-1)$ matrix $\mathbf{Y}_{-i}$ refers to $\mathbf{Y}$ excluding the ith column. $(p-1) \times 1$ vector $\mathbf{b}_i$ is the ith column of matrix $\mathbf{B}$ excluding the ith entry whose value has been already known to be zero. $q \times 1$ vector $\mathbf{f}_i$ denotes the ith column of $\mathbf{F}$. $\mathbf{e}_i$ is an $N \times 1$ error vector, we assume elements in $\mathbf{e}_i$ are independent and identical distributed as $N(0, \sigma^2)$, that is a normal distribution with 0 means and variance σ^2.

Without any other restrictions, the task of inferring a GRN from such a model is to estimate the $p(p-1)$ unknown entries in $\mathbf{B}$ and passingly estimate the pq unknown entries in $\mathbf{F}$. Since the GRNs or more general biochemical networks are always sparse [22], [30], the adjacency matrix $\mathbf{B}$ is sparse. We assume that the loci of *cis*-eQTLs have been obtained by

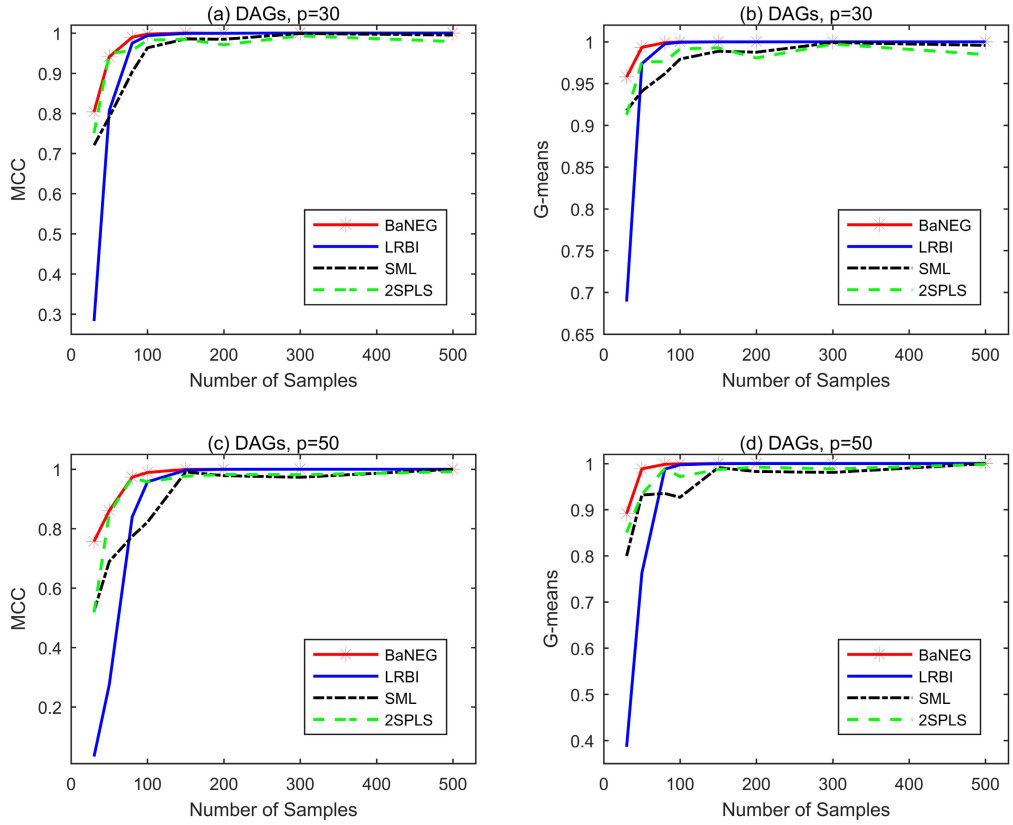

FIGURE 2. MCC and G-means of BaNEG, LRBI, SML and 2SPLS for DAGs with 30 genes and 50 genes.

applying existing eQTL mapping methods, whereas the effective sizes are unknown. As elucidated in [44], most eQTLs have weak effects on genes expression levels, so matrix $\mathbf{F}$ is usually a sparse matrix having a small number of nonzero entries whose locations have been determined. Despite both $\mathbf{B}$ and $\mathbf{F}$ are sparse, for relatively large p, the inference task is still challenging.

B. MODEL REPARAMETERIZATION

According to the rule of matrix multiplication, the SEM in (1) can be rewritten as the following form,

$$\mathbf{Y} = \mathbf{W}\boldsymbol{\beta} + \mathbf{E}, \tag{3}$$

where $\mathbf{W} = [\mathbf{Y}, \mathbf{X}]$ is an $N \times (p+q)$ design matrix, and $\boldsymbol{\beta} = [\mathbf{B}, \mathbf{F}]^T$ is the parameter matrix composed of $(p+q) \times p$ parameters. Thus the original SEM is reparameterized as a multivariable linear model. In this linear model, the responding variable matrix $\mathbf{Y}$ is the same as that of the original SEM, the predictive variable matrix $\mathbf{W}$ is an $N \times (p+q)$ matrix including both of the exogenous and endogenous variables. Therefore, our main concern becomes the $(p+q) \times p$ parameters in matrix $\boldsymbol{\beta}$.

Model (3) can also be easily split into p univariable linear models as follows,

$$\mathbf{y}_i = \mathbf{W}\boldsymbol{\beta}_i + \mathbf{e}_i, \quad i = 1, \cdots, p, \tag{4}$$

where $\boldsymbol{\beta}_i$ is the ith column of $\boldsymbol{\beta}$, namely a $(p+q) \times 1$ vector. As aforementioned, $\mathbf{B}$ and $\mathbf{F}$ are both sparse. What's more, the diagonal elements of $\mathbf{B}$ are zeros, and the locations of nonzero elements in $\mathbf{F}$ have been obtained in advance, that is, the row indices of the unknown parameters need to be estimated are known before the inference. We adopt a vector $\mathbf{s}_i$ to denote the row indices of the parameters to be estimated in $\boldsymbol{\beta}_i$. By selecting the columns of design matrix $\mathbf{W}$ identified by $\mathbf{s}_i$, the model (4) can be simplified into

$$\mathbf{y}_i = \mathbf{W}_{\mathbf{s}_i}\boldsymbol{\beta}_{\mathbf{s}_i} + \mathbf{e}_i, \quad i = 1, \cdots, p, \tag{5}$$

where $\mathbf{W}_{\mathbf{s}_i}$ refers to a reduced form of $\mathbf{W}$ that only contains the columns in accordance with $\mathbf{s}_i$, and $\boldsymbol{\beta}_{\mathbf{s}_i}$ is a reduced parameter vector that only includes the unknown rows. The dimensions of $\boldsymbol{\beta}_{\mathbf{s}_i}$ for different values of i may be different because our BaNEG algorithm allows different number of *cis*-eQTLs for each gene. We use p_i to represent the dimension of $\boldsymbol{\beta}_{\mathbf{s}_i}$. As such, our task is transformed into p linear models as shown in (5), in which the coefficients vectors are known to be sparse.

C. THE BAYESIAN LASSO

The most intuitive approaches to solve a sparse linear regression model is lasso [46] and its extensions such as SCAD [47], Elastic net [48], fused lasso [49], adaptive lasso [50], Bayesian-type lasso [51]–[53].

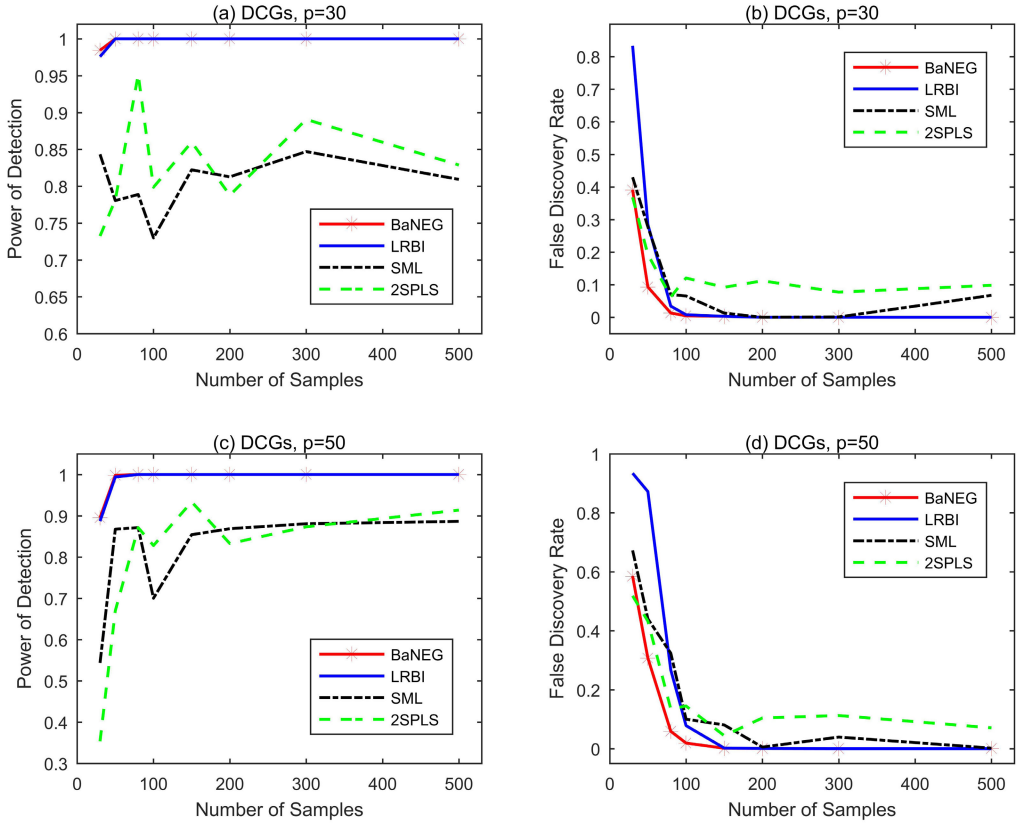

FIGURE 3. PD and FDR of BaNEG, LRBI, SML and 2SPLS for DCGs with 30 genes and 50 genes.

In Bayesian frameworks, lasso estimates can be interpreted as posterior mode estimates with independent and identical Laplace prior for coefficients [46], [51]. A Laplace prior for $\boldsymbol{\beta}_{\mathbf{s}_i}$ in (5) can be represented as a scaled mixture of normals [54]:

$$\begin{aligned}\pi\left(\boldsymbol{\beta}_{\mathbf{s}_i}|\sigma^2\right) &= \prod_{j=1}^{p_i} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\lambda|\beta_{\mathbf{s}_i,j}|/\sqrt{\sigma^2}} \\ &= \prod_{j=1}^{p_i} \mathrm{N}(\beta_{\mathbf{s}_i,j}|0, \sigma^2\tau_j^2)\mathrm{Exp}(\tau_j^2|\frac{\lambda^2}{2})d\tau^2. \quad (6)\end{aligned}$$

In the above Bayesian lasso prior, $\mathrm{Exp}(\tau_j^2|\frac{\lambda^2}{2})$ represents exponential distribution with rate parameter $\frac{\lambda^2}{2}$, where λ^2 is the penalty parameter that encourage the shrinkage of $\beta_{\mathbf{s}_i,j}$. Any Inverse-Gamma$(\nu_0/2, \eta_0/2)$ prior for σ^2 can maintain conjugacy, for convenient calculation, an improper prior density $\pi(\sigma^2) = 1/\sigma^2$ can also be used to model the error variance [51].

Since $\mathbf{e}_i$ is assumed to follow a normal distribution with zero mean and variance σ^2, the likelihood of model (5) can be expressed as

$$\mathbf{y}_i|\mathbf{W}_{\mathbf{s}_i}, \boldsymbol{\beta}_{\mathbf{s}_i}, \sigma^2 \sim \mathrm{N}_N(\mathbf{W}_{\mathbf{s}_i}\boldsymbol{\beta}_{\mathbf{s}_i}, \sigma^2\mathbf{I}_N) \quad (7)$$

According to Bayes' theorem, the joint posterior distribution can be obtained via

$$\pi(\boldsymbol{\beta}_{\mathbf{s}_i}|\mathbf{y}_i) \propto f(\mathbf{y}_i|\boldsymbol{\beta}_{\mathbf{s}_i})\pi(\boldsymbol{\beta}_{\mathbf{s}_i}|\sigma^2)\pi(\sigma^2). \quad (8)$$

D. INFER GRNS VIA BAYESIAN ADAPTIVE LASSO WITH NEG PRIOR

One of the approaches to selecting λ^2 is to give λ^2 a hyper-prior of Gamma distribution [51], which motivating the following three-level NEG prior [55]:

$$\begin{aligned}&\boldsymbol{\beta}_{\mathbf{s}_i}|\sigma^2, \tau_1^2, \cdots, \tau_{p_i}^2 \sim \mathrm{N}_{p_i}(\mathbf{0}_{p_i}, \sigma^2\mathbf{D}_{\boldsymbol{\tau}}), \\ &\tau_1^2, \cdots, \tau_{p_i}^2|\psi_j \sim \prod_{j=1}^{p_i} \mathrm{Exp}(\psi_j), \\ &\psi_j|a, b \sim \mathrm{Gamma}(a_0, b_0), \quad a_0 > 0,\ b_0 > 0, \quad (9)\end{aligned}$$

where $\mathbf{D}_{\boldsymbol{\tau}} = \mathrm{diag}(\tau_1^2, \cdots, \tau_{p_i}^2)$, and ψ_j is the penalty parameter corresponding to λ^2 in the Bayesian lasso. Note that in the Bayesian lasso, the same penalty parameter λ^2 for all coefficients in each iteration. However, generally we tend to put larger penalty parameters on coefficients corresponding to less important variables, which can increase shrinkage degree of such coefficients. So we adopt the Bayesian adaptive lasso to put different penalty parameters ψ_j on each coefficient $\beta_{\mathbf{s}_i,j}$. In the third level of the NEG prior in (9), a conjugate Gamma prior with a shape parameter a_0 and an inverse scale

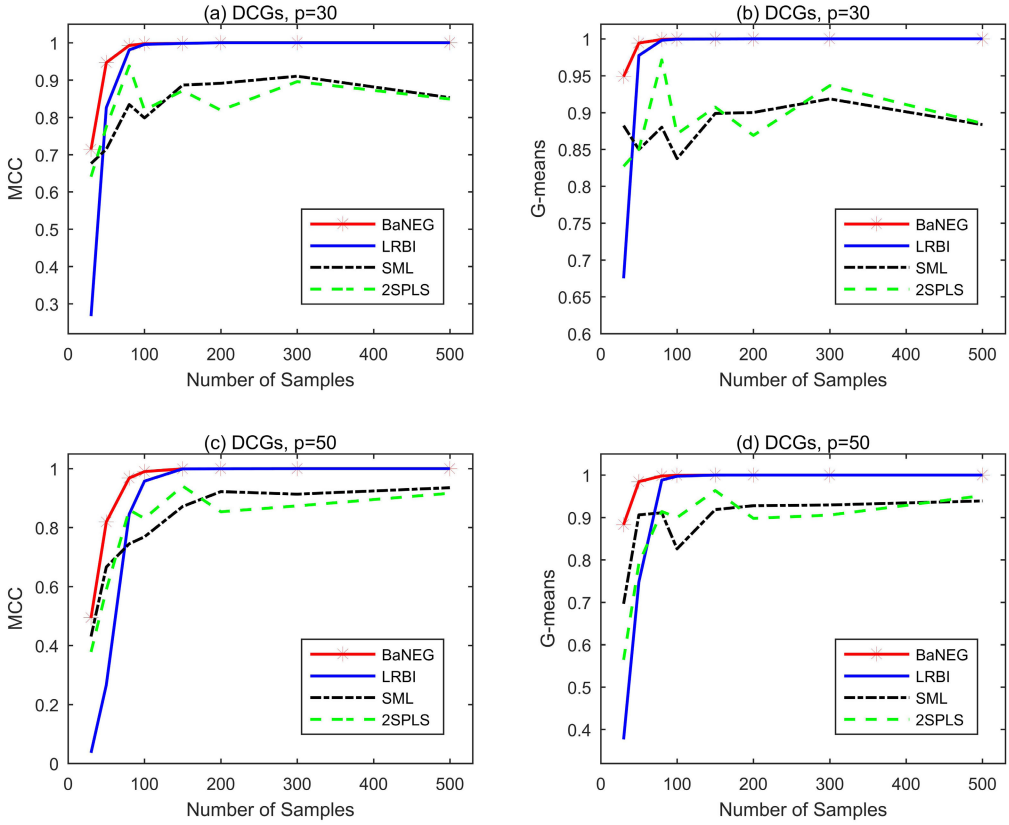

FIGURE 4. MCC and G-means of BaNEG, LRBI, SML and 2SPLS for DAGs with 30 genes and 50 genes.

parameter b_0 are assigned to ψ_j, which alleviates the strong dependence of shrinkage degree on ψ_j. Actually, we do not need to pay much attention on the choice of hyper parameters ν_0, η_0, a_0, b_0, because the deeper parameters are in the hierarchical Bayesian model, the less effects they have on the inference [56]. So appropriate values can be pre-specified for these hyper parameters.

By assigning σ^2 an Inverse-Gamma prior with shape parameter $\nu_0/2$ and inverse scale parameter $\eta_0/2$, the full-conditional posterior distribution of all the parameters $(\boldsymbol{\beta}_{\mathbf{s}_i}, \sigma^2, 1/\tau_j^2, \psi_j), j = 1, \cdots, p_i$ can be given by

$$\begin{aligned}
\boldsymbol{\beta}_{\mathbf{s}_i}|\mathbf{y}_i, \mathbf{W}_{\mathbf{s}_i}, \sigma^2, \tau_1^2, \cdots, \tau_{p_i}^2 &\sim \mathrm{N}_{p_i}(A^{-1}\mathbf{W}_{\mathbf{s}_i}^T\mathbf{y}_i, \sigma^2 A^{-1}),\\
\sigma^2|\mathbf{y}_i, \mathbf{W}_{\mathbf{s}_i}, \boldsymbol{\beta}_{\mathbf{s}_i}, \tau_1^2, \cdots, \tau_{p_i}^2 &\sim \text{Inverse-Gamma}(\nu/2, \eta/2),\\
\frac{1}{\tau_j^2}|\beta_{\mathbf{s}_i,j}, \sigma^2, \psi_j &\sim \text{Inverse-Gauss}(\mu, \lambda),\\
\psi_j|a_0, b_0 &\sim \text{Gamma}(a, b),
\end{aligned} \tag{10}$$

where

$$\begin{aligned}
A &= \mathbf{W}_{\mathbf{s}_i}^T\mathbf{W}_{\mathbf{s}_i} + \mathbf{D}_{\boldsymbol{\tau}}^{-1},\\
\mathbf{D}_{\boldsymbol{\tau}} &= \text{diag}(\tau_1^2, \tau_2^2, \cdots, \tau_{p_i}^2),\\
\nu &= n + p_i + \nu_0,\\
\eta &= ||\mathbf{y}_i - \mathbf{W}_{\mathbf{s}_i}\boldsymbol{\beta}_{\mathbf{s}_i}||_2^2 + \boldsymbol{\beta}_{\mathbf{s}_i}^T\mathbf{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta}_{\mathbf{s}_i} + \eta_0,
\end{aligned}$$

$$\begin{aligned}
\mu &= \sqrt{\frac{2\psi_j\sigma^2}{\beta_{\mathbf{s}_i,j}^2}}, \quad \lambda = 2\psi_j,\\
a &= p_i + a_0, \quad b = \tau_j^2 + b_0.
\end{aligned} \tag{11}$$

It has been proved by Parl and Casella [51] and Leng *et al.* [52] in their Appendices that by assuming the above NEG type priors as in (9), the unimodal of posterior distribution can be guaranteed.

Then we can iteratively draw samples for all unknown parameters from the above full-conditional posterior distribution to estimate unknown parameters by using a Gibbs sampler. The convergence of the sampling process for such hierarchical models has been investigated and proved to be rapid [57]. To initialize the parameters, we preset $\nu_0 = N/5, \eta_0 = 1$. The hyper parameters a_0, b_0 are pre-specified as small values (e.g. $a_0 = 0.1, b_0 = 0.5$) to make the prior for ψ_j essentially noninformative. $\psi_j, \tau_j^2, \sigma^2$ are initialized by performing sampling from their prior distributions respectively. And the starting value of $\boldsymbol{\beta}_{\mathbf{s}_i}$ is set as a vector of 1's.

With the initialization of each parameter, the Gibbs sampler can be performed iteratively, all parameters $(\boldsymbol{\beta}_{\mathbf{s}_i}, \sigma^2, 1/\tau_j^2, \psi_j)$ are updated in turn by drawing samples from their conditional posterior distributions given current values of other parameters until sufficient effective samples of $\boldsymbol{\beta}_{\mathbf{s}_i}$ are obtained. We introduce the potential scale reduction

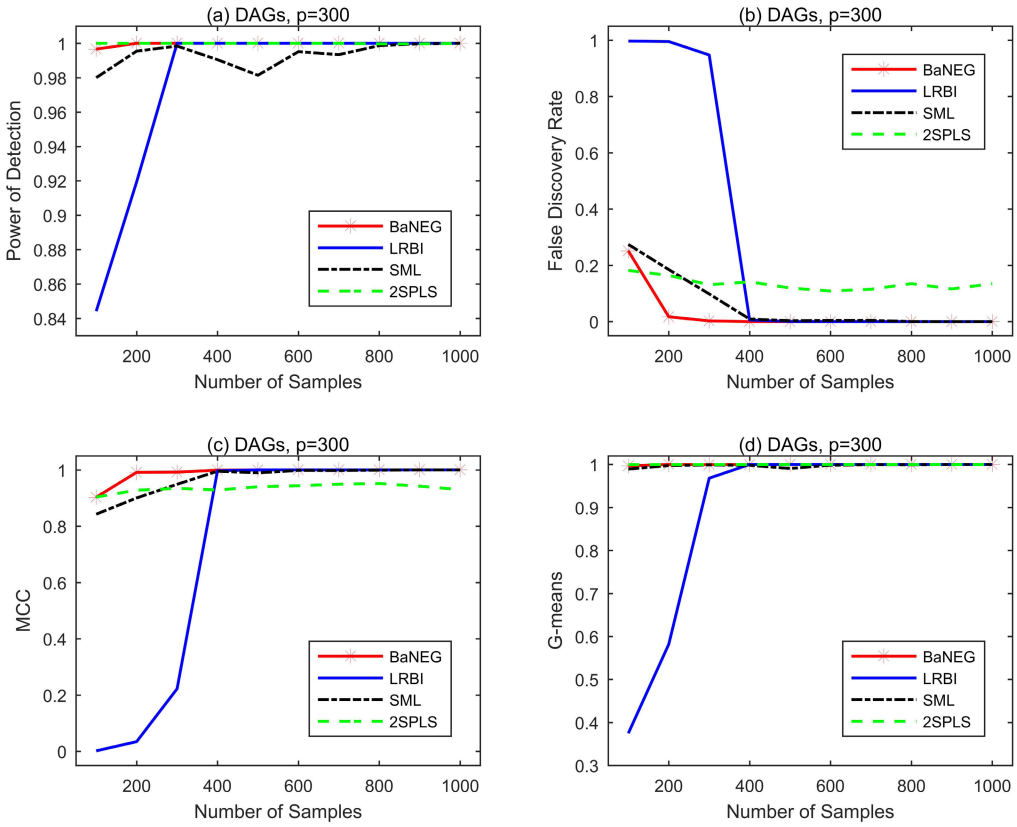

FIGURE 5. PD, FDR, MCC and G-means of BaNEG, LRBI, SML and 2SPLS for large DAGs with 300 genes.

factor $\widehat{R}$ [58] to monitor convergence of our Gibbs sampler. Two parallel chains with starting points set above are run, we calculate $\widehat{R}$ for all p_i entries of $\boldsymbol{\beta}_{\mathbf{s}_i}$ to compare between- and within- sequence variance. According to the description in [58], the convergence condition is set as $\widehat{R} < 1.1$. Once $\widehat{R}$ for the p_i entries of $\boldsymbol{\beta}_{\mathbf{s}_i}$ all meet the condition, the simulations can be identified convergent. The subsequent iterations generate effective samples. Two stopping conditions have been used in our simulation as well to avoid excessive number of iterations. First, a maximum number of iterations is set (usually is set as 50 in the following). Second, after each effective iteration, we compare our interested parameters $\boldsymbol{\beta}_{\mathbf{s}_i}$ with that of last iteration, once the sum of all square differences is small enough, we think the iteration has reached a stable state and can be stopped. For example, when k is smaller than the preset maximum number of iterations, in the $(k+1)$th iteration, we calculate and test if the value of $\sum_{j=1}^{p}(\beta_{\mathbf{s}_i,j}^{(k+1)} - \beta_{\mathbf{s}_i,j}^{(k)})^2$ is small than a pre-specified small threshold like 0.01, if so, we deem enough iterations have been simulated; otherwise, the sampling iterations need to be performed continuously. Then we discard half of the convergent posterior samples by only extracting every 2nd simulation drawn from each sequence to avoid dependence between two adjacent iterations. From the simulations in next section, we find that with this sampling strategy, the Gibbs sampler can achieve convergence fast.

Different from non-Bayesian lasso, the penalized analysis in Bayesian frameworks do not shrink the insignificant coefficients to zero automatically. They just shrink coefficients corresponding to insignificant variables to very small values. The simplest way for variable selection is to apply a threshold *thr*, more specifically, if an estimated coefficient $\beta_{\mathbf{s}_i,j}$ is larger than the pre-specified threshold *thr*, it remains unchanged; otherwise, it is shrunken to zero. The smaller *thr* is, the performance in terms of PD would be better, meanwhile the FDR would be accordingly worse; conversely, a larger *thr* produce better FDR but worse PD.

III. RESULTS

A. SIMULATION STUDIES

The performance of a GRN inference algorithm is often measured by PD and FDR. PD is equivalent to the true positive rate, which measures the proportion of true edges that are correctly identified in all true edges. FDR measures the proportion of false positive edges in all detected edges. Assuming a positive edge and a negative edge indicate an edge exists or not respectively. Let N_{tp} denote the number of true positive edges detected by an inference algorithm, N_{fp} stand for the number of false positive edges, N_{tn} represent the number of true negative edges, and N_{fn} characterize the number of false negative edges. Then PD can be obtained by $N_{tp}/(N_{tp}+N_{fn})$, the FDR can be calculated from $N_{fp}/(N_{fp}+N_{tp})$.

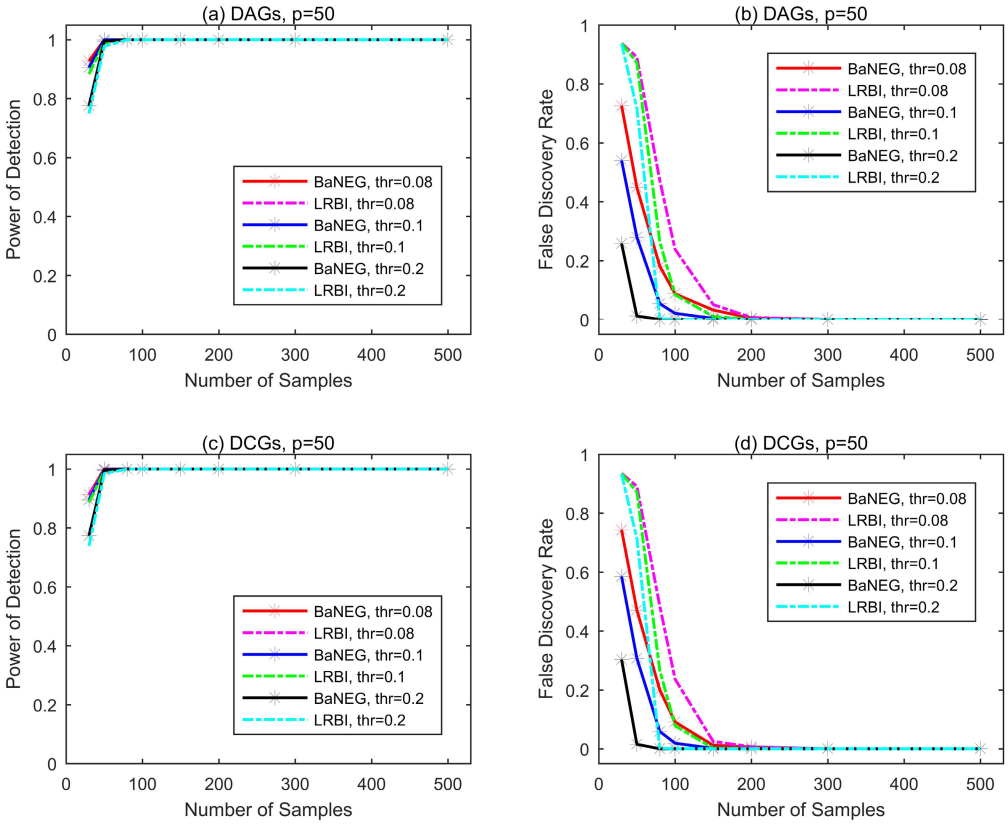

FIGURE 6. Performance of BaNEG and LRBI for DAGs and DCGs with different values of *thr*.

In the following simulations, we mainly evaluate the performance of the GRN inference algorithms by comparing their PD and FDR. Our main goal is to result in higher PD and lower FDR, the ideal situation is PD $=1$ and FDR $=0$. In addition, several other measures are also taken into considerations to further compare other performance of the algorithms, including Matthews Correlation Coefficient (MCC) and G-mean. MCC is a correlation coefficient (ranging from -1 to 1) evaluating the correlation between a detected GRN and the corresponding true GRN, and G-mean (ranging from 0 to 1) is usually used to evaluate performance for imbalanced data. The bigger the values of these two measures are, the better performance an algorithm has (the best situation is MCC $=1$ and G-mean $=1$).

To benchmark the performance of our BaNEG algorithm, we compare its performance with some other similar algorithms that infer GRNs modeled with SEMs incorporating genetic perturbations into models. A series of algorithms have been proposed to solve such problem, such as the PC-algorithm [59], the QDG algorithm [34], the QTLnet algorithm [35], the NEO algorithm [60], the AL-based algorithm [39], the SML algorithm [40], the LRBI algorithm [42] and the 2SPLS algorithm [43]. Logsdon et al. have compared the performance of their AL-based algorithm with that of the NEO algorithm, the QDG algorithm, the QTLnet algorithm and the PC-algorithm, the simulation results in [39] demonstrate that AL generally outperforms the other four algorithms. Later, Cai et al. compared their SML algorithm with the QDG algorithm and the AL-based algorithm in [40], the results showed that SML has significantly better performance than these two algorithms. Combined with the further comparative simulations in [42], [43], we can draw a conclusion that SML and 2SPLS outperform all other algorithms listed above in terms of FDR, while LRBI and 2SPLS perform the best in terms of PD. Therefore, in this section, we run simulations on synthetic data to compare the performance of BaNEG with LRBI, SML and 2SPLS to prove the superiority of BaNEG.

The GRNs and corresponding synthetic data used in this section are simulated following the way as recommended in [39], [40], [42]. The structure of a simulated GRN can be represented by a $p \times p$ adjacency matrix $\mathbf{B}$, if there is an edge from node i to node j, the regulation effects B_{ij} are randomly generated from a uniform distribution over the interval $(-1,-0.5)\bigcup(0.5,1)$. Other elements and all diagonal terms of $\mathbf{B}$ are set to 0. For convenient calculation, without loss of generality, we assume $\mathbf{F}$ a $p \times p$ identify matrix, that is each gene has and only has one cis-eQTL. So the genotype matrix $\mathbf{X}$ is an $N \times p$ matrix. The genotype of each eQTL X_{ij} is randomly generated following the setting of F2 cross,

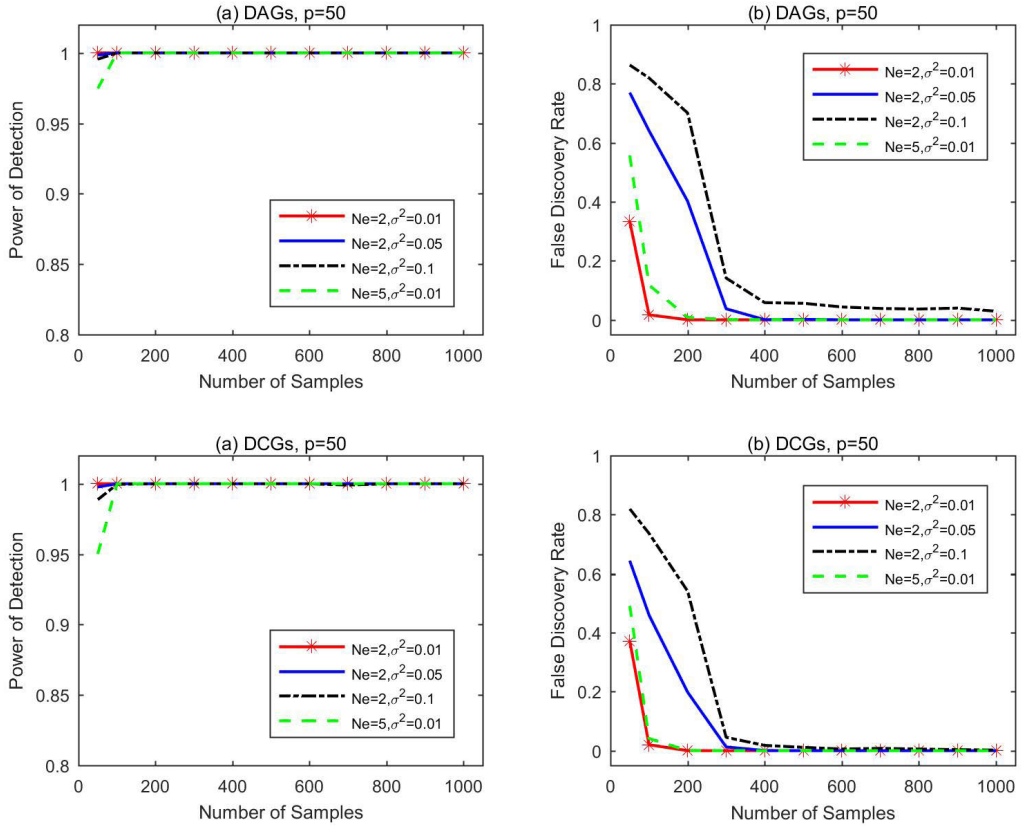

FIGURE 7. Performance of BaNEG algorithms for DAGs and DCGs with different N_e and σ^2.

that is, takes value of 1, 2, 3 with probabilities 0.25, 0.5, 0.25, respectively. Each error term E_{ij} is simulated from a Gaussian distribution with zero mean and variance σ^2 and intercept term is set to zero. Then $\mathbf{Y}$ can be obtained from $\mathbf{Y} = (\mathbf{XF} + \mathbf{E})(\mathbf{I} - \mathbf{B})^{-1}$.

Firstly, we conduct simulation studies to compare performance of our BaNEG algorithm with LRBI [42], SML [40] and 2SPLS [43]. DAGs and DCGs composed of 30 genes and 50 genes are simulated with $\sigma^2 = 0.01$. Each node is simulated to have $N_e = 3$ regulatory edges on average. The sample size N ranges from 30 to 500. The variable selection threshold *thr* for BaNEG and LRBI are set to 0.1. For each type of GRN with 30 genes, We simulate 100 replicates, and for larger GRNs with 50 genes, 50 replicates are generated. Performance of each algorithm is obtained by averaging the PD and FDR of all replicates in the same setup. Simulation results of DAGs with 30 genes and 50 genes are presented in Fig. 1 and Fig. 2, and the results of DCGs are shown in Fig.3 and Fig.4. From Fig. 1 (a) (c), we can see for DAGs, the PD of BaNEG and LRBI are close to or equal to 1 for sample sizes from 50 to 500, BaNEG has very slightly better PD than LRBI when $N = 30$ and 50. Whereas the PD of SML and 2SPLS are obviously lower than that of BaNEG. As shown in Fig. 1 (b) (d), for DAGs, the FDR of BaNEG stays at 0 for sample sizes from 150 to 500, and is lower than that of other three algorithms for almost all sample sizes. The only exception case is when $N = 30$, the FDR of BaNEG is larger than 2SPLS, but at the same time, the PD of 2SPLS is much lower than BaNEG. As for MCC and G-means shown in Fig. 2, BaNEG performs better than all of the other three algorithms visibly. Fig. 3 and Fig. 4 present PD, FDR, MCC and G-means of the four algorithms for DCGs, where we can find that the performance of BaNEG and LRBI are similar with that for DAGs, but SML and 2SPLS perform obviously worse than that for DAGs, meaning that the superiority of BaNEG over SML and 2SPLS are more remarkable.

Secondly, we continue to evaluate and compare the performance of the above four algorithms for larger sparse DAGs with 300 genes, here we set $N_e = 1$. The noise variances σ^2 are still set to 0.01, the sample sizes are from 100 to 1000 and 10 replicates are simulated for each sample size. The performance of BaNEG, LRBI, SML and 2SPLS are shown in Fig. 5. From Fig. 5 (a) (b), the PD of BaNEG are euqal to 1 except for the case when sample size $N = 100$, which outperforms that of LRBI, SML and 2SPLS for almost all sample sizes; the FDR of BaNEG are exactly equal to 0 for sample sizes from 400 to 1000, which also performs better than all the other three algorithms. As shown in Fig. 5 (c) (d), the MCC and G-means of BaNEG are equal to 1 except very few cases, whereas other three algorithms all perform weaker

TABLE 1. The running time (in seconds) of BaNEG, LRBI, SML and 2SPLS for DAGs and DCGs with 30, 50, 100 and 300 genes at sample size $N = 500$.

	DAGs				DCGs			
	p=30	p=50	p=100	p=300	p=30	p=50	p=100	p=300
BaNEG	10.5	21.3	112.3	1104.8	10.8	22.1	109.1	1034.5
LRBI	0.8	2.2	8.4	92.5	0.7	1.9	9.5	88.9
SML	68.7	101.8	246.9	4065.9	96.1	171.6	2774.7	3963.4
2SPLS	28.5	57.5	209.3	4737.5	25.1	91.0	505.7	4646.5

than BaNEG. Note that LRBI shows rather poor performance for small sample sizes (like $N < 400$).

Then, we run simulations on synthetic data sets with $p = 50, N_e = 3, \sigma^2 = 0.01$ to study the impact of different variable selection threshold *thr* on BaNEG. DAGs and DCGs are simulated with *thr* ranging in $\{0.08, 0.1, 0.2\}$, and the sample sizes are from 30 to 500. Aside from BaNEG based on Bayesian adaptive lasso with NEG prior, the other Bayesian based method LRBI is also applied on the simulated data sets. The simulation results are presented in Fig. 6. From Fig. 6 (a) (c), we can find that the PD of the two algorithms are all equal to 1 for sample sizes from 100 to 500 with all different values of *thr*, nevertheless, when $N = 30$ or 50, the PD of BaNEG slightly exceeds that of LRBI with the same *thr*. The difference of FDR as shown in Fig. 6 (b) (d) is relatively more distinct. In an overall view, the FDR of BaNEG outperforms that of LRBI, especially for small sample sizes, and the FDR of each algorithm with a larger *thr* is better than that with a lower *thr*.

Moreover, we run simulations on DAGs and DCGs with 50 genes to analyze the performance of BaNEG for different N_e and σ^2. We continue to set $thr = 0.1$ for all simulations. DAGs and DCGs are simulated with N_e equaling to 2 or 5 and σ^2 ranging in $\{0.01, 0.05, 0.1\}$. The simulation results for sample sizes from 50 to 1000 are depicted in Fig. 7. As shown in Fig. 5, when we keep N_e constant and increase σ^2 from 0.01 to 0.05 and 0.1, the PD of both DAGs and DCGs suffer little affection, only reduced a little bit for sample size $N = 50$, whereas the performance in terms of FDR are negatively impacted evidently. When we keep σ^2 constant and increase N_e from 2 to 5, both of the PD and FDR become slightly worse.

Finally, to compare the computational expenses of the above four algorithms, we record the running time of each algorithm when infer DAGs and DCGs with 30, 50, 100, 300 genes from the same data sets. All the algorithms were conducted by using a laptop with Intel(R) Core(TM) i7-6700HQ CPU 2.60GHz and 16G RAM. Reported in Table 1 are the running times of all algorithms on different GRNs at sample size $N = 500$. As shown in Table 1, while maintaining the performance advantage of BaNEG over the other three algorithms, BaNEG is much faster than SML and 2SPLS. However, it is slower than LRBI visibly because its more complicated hierarchical posterior model and convergence monitor strategy. According to the previous comparisons in various of performance measures, such moderate sacrifice in computational cost brought obvious advantages in network accuracy.

B. REAL DATA ANALYSIS

In this section, we apply the BaNEG algorithm to a real data set composed of 47 yeast genes from 112 segregates of Saccharomyces cerevisiae to explore the corresponding underlying GRNs. To further verify the simulation results in last section, we also run the other three algorithms: LRBI, SML, 2SPLS on the real data set.

Brem and Kruglyak [44] measured expression levels of 5727 genes and genotypes of 2957 genetic markers from 112 segregates of a cross between a lab strain BY4716 and a wild strain RM11-1a of Saccharomyces cerevisiae. The sample size is too small compared with the number of genes. Chen and Ren [43] filtered the data set and yielded a data set with 112 observations of 722 genes and 732 genotypes of *cis*-eQTLs. The 2SPLS algorithm was applied to infer this big GRN and generates a GRN with 323 edges, which formed a few subnetworks. The three largest subnetworks consist of 47 genes and 60 regulatory edges in total. We apply our proposed BaNEG algorithm to the yeast data set including gene expression levels and genotypes of *cis*-eQTLs of the 47 genes from 112 samples to infer the underlying yeast GRN. In the data set, the gene expression matrix is corresponding to the matrix $\mathbf{Y}$ in model (1) and the *cis*-eQTLs data represents the matrix $\mathbf{X}$, the BaNEG algorithm can be directly applied to the data sets to infer the adjacency matrix $\mathbf{B}$, thereby construct the underlying GRN of the 47 yeast genes.

As shown in Fig. 6, when we increase *thr* from 0.1 or 0.2, the FDR of BaNEG can be greatly improved, the PD became slightly lower but did not suffered too much influences. So when we make real data analysis on the yeast data set of 47 genes with 112 observations, we set $thr = 0.2$ to lower the FDR as far as possible and maintain good-enough PD in the meanwhile. In this setting, 324 regulatory edges were detected when BaNEG was applied. To improve the reliability of the inferred GRN, 100 bootstrap data sets with 112 samples were generated by random re-sampling the original 112 observations with replacement. Then the BaNEG algorithm was run on each bootstrap data set and a frequency matrix was constructed by counting the frequency of each edges in the 100 inferred networks. By eliminating the edges whose corresponding frequency were less than 80, we obtained a GRN of 45 genes and 90 edges, including 69 positive effects and 21 negative effects. The inferred GRN is shown in Fig. 8, each node stands for a gene and a directed edge from node $\mathcal{A}$ to node $\mathcal{B}$ indicates the source gene $\mathcal{A}$ has a regulatory effect on the target gene $\mathcal{B}$, a positive effect in the adjacency matrix denotes an activating effect (depicted as $\rightarrow$) and a negative effect stands for an inhibiting effect (depicted as $\dashv$). Most genes were detected at least one regulatory or regulated edge. Only 2 genes, namely YJL140W, YPL031C, were excluded from the inferred GRN. There are

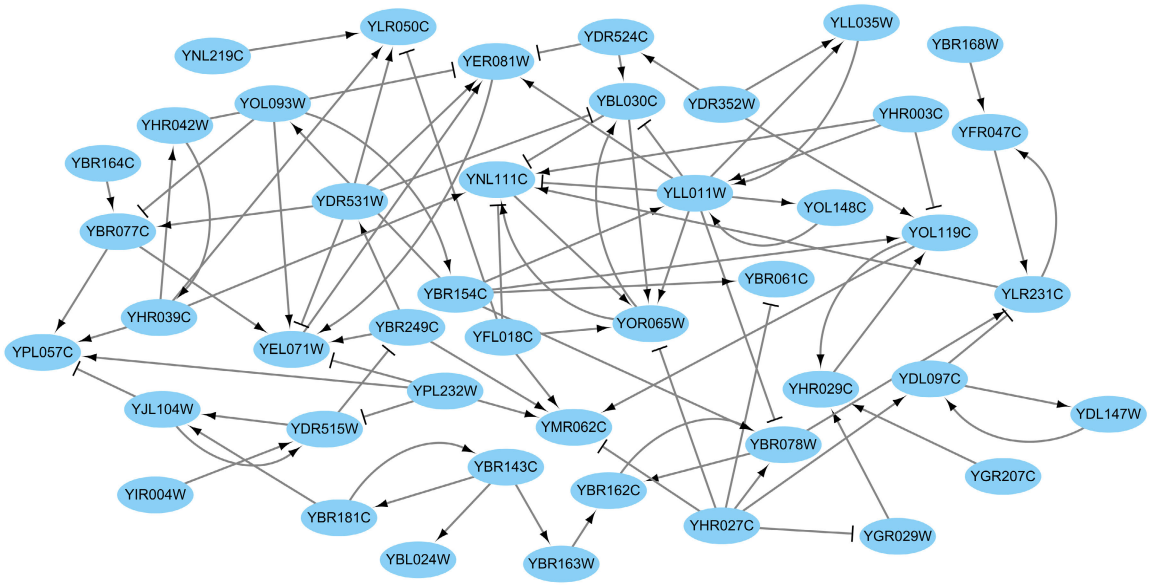

FIGURE 8. The GRN of 47 yeast genes inferred from gene expression data and eQTL data by applying the BaNEG algorithm.

11 pairs of genes have mutual regulatory effects and all of them are activating effects.

According to the simulation results in last section, the BaNEG algorithm has similar PD with LRBI, larger PD than SML and 2SPLS, and has lower FDR than all of the other three algorithms. So we expect that for the same real data set, the BaNEG algorithm can detect more true edges than SML and 2SPLS, and may detect less false edges than the LRBI algorithm. Then LRBI, SML and 2SPLS were also run with 100 bootstraps, and only the edges whose frequencies were not less than 80 were retained in the inferred GRNs. As a result, the LRBI algorithm detected 46 genes and 155 edges, the only eliminated gene is YPL031C, which was also not included in the GRN inferred by BaNEG. Besides, almost all edges detected by BaNEG are included in the GRN identified by LRBI (only 5 edges were not in). The SML algorithm only found 30 genes and 37 edges, in which 27 edges were also in the BaNEG GRN. And the GRN inferred by the 2SPLS algorithm included 42 genes and 58 edges, 26 edges were in the BaNEG GRN. By comparing the results of the four algorithms inferred from the yeast data set, we found that our BaNEG algorithm detected less edges than LRBI and more edges than SML and 2SPLS due to its lower FDR and higher PD, which is in accordance with our expectation based on the simulation studies.

IV. DISCUSSION

Since the development of high-throughput sequencing technologies, considerable efforts were made to infer GRNs from gene expression data [17], [20], [23], [25], [30]. While most of these traditional methods were developed to deal with only the gene expression data. Another type of approaches were developed in recent years to integrated genetic perturbations with gene expression data to together infer GRNs, by exploiting additional genetic information, the accuracy of inference can be improved. SEMs provide a systematic framework that can directly integrate both types of gene data and offer flexibility to model both DAGs and DCGs by adjacency matrices [40]. Motivated by this, in this paper, we develop a more efficient novel approach based on Bayesian learning named BaNEG to infer GRNs modeled with SEMs.

In BaNEG, we combine the three level NEG type prior with Bayesian lasso to form an adaptive hierarchical posterior model for sparse linear models, then apply it to SEMs to infer GRNs from both gene expression data and gene perturbations for the first time. It is realized by two stages: First, the original SEM is reparameterized as a linear type model by merging the endogenous variables and the exogenous variables; then we propose to use Bayesian adaptive lasso with NEG prior and Gibbs sampling to infer the reparameterized linear type model. This proposed algorithm mainly has the following advantages: First, the reparameterization stage transfers SEMs to linear models. This linear model exploits the whole structure of SEM by integrating gene expression level and *cis*-eQTL loci into one design matrix, which makes it possible to infer the parameters together. Second, the simplified form as in model (5) reduces the dimension of the re-parameterized models, this significantly improves the inference efficiency, and would be more applicable for large GRNs (such as whole-genome GRNs). Finally, BaNEG adopts NEG type prior to achieve sparsity of GRNs. In another Bayesian based algorithm LRBI, an NG-type prior was chosen, which may have an infinite spike at zero and flatness for large values of coefficients, as a consequence, does not penalize such large values. Therefore, with the NG-type prior, the spike at zero has strong consequences for the model behavior of the posterior, not all of which are welcome. While the NEG

distribution incorporates as limiting cases of the Laplace prior and has the advantage of a finite limit at zero for all parameter values in its range [55]. The simulations have confirmed that BaNEG maintains the high PD of LRBI (even be slightly better), and meanwhile receives much lower FDR.

For the analysis of a real data set including a yeast data set with 47 genes from 112 samples, BaNEG discovered more potential edges than the SML and the 2SPLS but less edges than LRBI. Combined with the simulation studies on synthetic data, there are good reasons to believe that BaNEG discovered less false edges than LRBI, and detected more true edges than SML and 2SPLS. Hence, we believe that the GRN constructed by BaNEG is more reliable than other algorithms and is of great reference value for inference of GRNs, which is meaningful for discovering gene functions and gene-gene interactions.

V. CONCLUSION

The inference of GRNs is of significant and profound importance for better understanding the inherent biological mechanisms and precision medicine. In this study, we develop and present a method to infer the topology structures of GRNs from both gene expression data and *cis*-eQTL data. Systematic simulation studies demonstrate that the BaNEG algorithm outperforms three state-of-the-art algorithms (LRBI, SML and 2SPLS). The results inferred from a real data set supports the simulation results and therefore can be considered reasonable and meaningful in a biological sense. In conclusion, the BaNEG algorithm is considered to be an effective and efficient approach that can be used to infer underlying GRNs from gene expression data and genetic perturbations, and would be a useful tool for medical treatment and genetic research in practical.

REFERENCES

[1] G. C. Karp, "Introduction to the study of cell and molecular biology," in *Cell and Molecular Biology: Concepts and Experiments* 6th ed. Hoboken, NJ, USA: Wiley, 2009, pp. 1–24.
[2] V. Emilsson and G. Thorleifsson, "Genetics of gene expression and its effect on disease," *Nature*, vol. 452, no. 7186, pp. 423–428, Mar. 2008.
[3] S. Chu, "The transcriptional program of sporulation in budding yeast," *Science*, vol. 282, no. 5389, pp. 699–705, Oct. 1998.
[4] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown, "Genomic expression programs in the response of yeast cells to environmental changes," *Mol. Biol. Cell*, vol. 11, no. 12, pp. 4241–4257, Dec. 2000.
[5] T. R. Hughes *et al.*, "Functional discovery via a compendium of expression profiles," *Cell*, vol. 102, no. 1, pp. 109–126, Jul. 2000.
[6] P. Brazhnik, A. de la Fuente, and P. Mendes, "Gene networks: How to put the function in genomics," *Trends Biotechnol.*, vol. 20, no. 11, pp. 467–472, Nov. 2002.
[7] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. di Bernardo, "How to infer gene networks from expression profiles," *Mol. Syst. Biol.*, vol. 3, no. 1, p. 122, Jun. 2007.
[8] N. Sun and H. Zhao, "Reconstructing transcriptional regulatory networks through genomics data," *Stat. Methods Med. Res.*, vol. 18, no. 6, pp. 595–617, Dec. 2009.
[9] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective heart disease prediction using hybrid machine learning techniques," *IEEE Access*, vol. 7, pp. 81542–81554, 2019.
[10] M. H. Bhatti, J. Khan, M. U. G. Khan, R. Iqbal, M. Aloqaily, Y. Jararweh, and B. Gupta, "Soft computing-based EEG classification by optimal feature selection and neural networks," *IEEE Trans Ind. Informat.*, vol. 15, no. 10, pp. 5747–5754, Oct. 2019.
[11] S. Oueida, Y. Kotb, M. Aloqaily, Y. Jararweh, and T. Baker, "An edge computing based smart healthcare framework for resource management," *Sensors*, vol. 18, no. 12, p. 4307, Dec. 2018.
[12] S. Jacob, V. G. Menon, F. Al-Turjman, V. P. G., and L. Mostarda, "Artificial muscle intelligence system with deep learning for post-stroke assistance and rehabilitation," *IEEE Access*, vol. 7, pp. 133463–133473, 2019.
[13] S. Oueida, M. Aloqaily, and S. Ionescu, "A smart healthcare reward model for resource allocation in smart city," *Multimedia Tools Appl.*, vol. 78, no. 17, pp. 24573–24594, Sep. 2018.
[14] D. R. Cacciagrano, F. Corradini, R. Culmone, N. Gorogiannis, L. Mostarda, F. Raimondi, and C. Vannucchi, "Analysis and verification of ECA rules in intelligent environments," *J. Ambient Intell. Smart Environ.*, vol. 10, no. 3, pp. 261–273, Jun. 2018.
[15] W. Fu, S. Liu, and G. Srivastava, "Optimization of big data scheduling in social networks," *Entropy*, vol. 21, no. 9, p. 902, Sep. 2019.
[16] F. Al-Turjman, H. Zahmatkesh, and L. Mostarda, "Quantifying uncertainty in Internet of medical Things and big-data services using intelligence and deep learning," *IEEE Access*, vol. 7, pp. 115749–115759, 2019.
[17] S. Liang, S. Fuhrman, and R. Somogyi, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," *Pacific Symp. Biocomput.*, vol. 3, pp. 18–29, Jan. 1998.
[18] T. Akutsu, S. Miyano, and S. Kuhara, "Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function," *J. Comput. Biol.*, vol. 7, nos. 3–4, pp. 331–343, Aug. 2000.
[19] H. Lahdesmaki, I. Shmulevich, and O. Yli-Harja, "On Learning Gene Regulatory Networks Under the Boolean Network Model," *Mach. Learn.*, vol. 52, nos. 1–2, pp. 147–167, Jul. 2003.
[20] W. Luo, K. D. Hankenson, and P. J. Woolf, "Learning transcriptional regulatory networks from high throughput gene expression data using continuous three-way mutual information," *BMC Bioinf.*, vol. 9, no. 1, p. 467, 2008.
[21] X. Zhang, X.-M. Zhao, K. He, L. Lu, Y. Cao, J. Liu, J.-K. Hao, Z.-P. Liu, and L. Chen, "Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information," *Bioinformatics*, vol. 28, no. 1, pp. 98–104, Nov. 2011.
[22] A. Villaverde, J. Ross, and J. Banga, "Reverse engineering cellular networks with information theoretic methods," *Cells*, vol. 2, no. 2, pp. 306–329, May 2013.
[23] B. Li, H. Chun, and H. Zhao, "Sparse estimation of conditional graphical models with application to gene networks," *J. Amer. Stat. Assoc.*, vol. 107, no. 497, pp. 152–167, Mar. 2012.
[24] J. H. Oh and J. O. Deasy, "Inference of radio-responsive gene regulatory networks using the graphical lasso algorithm," *BMC Bioinf.*, vol. 15, no. S7, May 2014.
[25] M. Zou and S. D. Conzen, "A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data," *Bioinformatics*, vol. 21, no. 1, pp. 71–79, Aug. 2004.
[26] M. Wang, Z. Chen, and S. Cloutier, "A hybrid Bayesian network learning method for constructing gene networks," *Comput. Biol. Chem.*, vol. 31, nos. 5–6, pp. 361–372, Oct. 2007.
[27] Y. Zhang and Z. Deng, "Inferring gene regulatory networks from multiple data sources via a dynamic Bayesian network with structural EM," in *Proc. Int. Conf. Data Integr. Life Sci.*, Philadelphia, PA, USA, 2007, pp. 204–214.
[28] N. Sene and G. Srivastava, "Generalized mittag-leffler input stability of the fractional differential equations," *Symmetry*, vol. 11, no. 5, p. 608, May 2019.
[29] L. F. Iglesias-Martinez, W. Kolch, and T. Santra, "BGRMI: A method for inferring gene regulatory networks from time-course gene expression data and its application in breast cancer research," *Sci. Rep.*, vol. 6, no. 1, Nov. 2016, Art. no. 37140.
[30] T. S. Gardner, "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, no. 5629, pp. 102–105, Jul. 2003.
[31] S. Basu, K. Kumbier, J. B. Brown, and B. Yu, "Iterative random forests to discover predictive and stable high-order interactions," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 8, pp. 1943–1948, Jan. 2018.

[32] J. Zhu, M. C. Wiener, C. Zhang, A. Fridman, E. Minch, P. Y. Lum, J. R. Sachs, and E. E. Schadt, "Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations," *PLoS Comput. Biol.*, vol. 3, no. 4, p. e69, 2007.

[33] J. Zhu, B. Zhang, E. N. Smith, B. Drees, R. B. Brem, L. Kruglyak, R. E. Bumgarner, and E. E. Schadt, "Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks," *Nature Genet.*, vol. 40, no. 7, pp. 854–861, Jun. 2008.

[34] E. Chaibub Neto, C. T. Ferrara, A. D. Attie, and B. S. Yandell, "Inferring causal phenotype networks from segregating populations," *Genetics*, vol. 179, no. 2, pp. 1089–1100, May 2008.

[35] E. Chaibub Neto, M. P. Keller, A. D. Attie, and B. S. Yandell, "Causal graphical models in systems genetics: A unified framework for joint inference of causal network and genetic architecture for correlated phenotypes," *Ann. Appl. Statist.*, vol. 4, no. 1, pp. 320–339, Mar. 2010.

[36] D. Thieffry, A. M. Huerta, E. Pérez-Rueda, and J. Collado-Vides, "From specific gene regulation to genomic networks: A global analysis of transcriptional regulation in escherichia coli," *BioEssays*, vol. 20, no. 5, pp. 433–440, Dec. 1998.

[37] J. Tegner, M. K. S. Yeung, J. Hasty, and J. J. Collins, "Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 10, pp. 5944–5949, May 2003.

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

[39] B. A. Logsdon and J. Mezey, "Gene expression network reconstruction by convex feature selection when incorporating genetic perturbations," *PLoS Comput. Biol.*, vol. 6, no. 12, Dec. 2010, Art. no. e1001014.

[40] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations," *PLoS Comput. Biol.*, vol. 9, no. 5, May 2013, Art. no. e1003068.

[41] 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, Feb. 2008.

[42] Z. Dong, T. Song, and C. Yuan, "Inference of gene regulatory networks from genetic perturbations with linear regression model," *PLoS ONE*, vol. 8, no. 12, Dec. 2013, Art. no. e83263.

[43] C. Chen and M. Ren, "Two-stage penalized least squares method for constructing large systems of structural equations," *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 40–73, 2018.

[44] R. B. Brem and L. Kruglyak, "The landscape of genetic complexity across 5,700 gene expression traits in yeast," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 5, pp. 1572–1577, Jan. 2005.

[45] B. E. Stranger, M. S. Forrest, M. Dunning, C. E. Ingle, C. Beazley, N. Thorne, R. Redon, C. P. Bird, A. de Grassi, C. Lee, C. Tyler-Smith, N. Carter, S. W. Scherer, S. Tavare, P. Deloukas, M. E. Hurles, and E. T. Dermitzakis, "Relative impact of nucleotide and copy number variation on gene expression phenotypes," *Science*, vol. 315, no. 5813, pp. 848–853, Feb. 2007.

[46] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B Methodol.*, vol. 58, no. 1, pp. 267–288, Dec. 2018.

[47] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *J. Amer. Stat. Assoc.*, vol. 96, no. 456, pp. 1348–1360, Dec. 2001.

[48] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc. B Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, Apr. 2005.

[49] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. Roy. Stat. Soc. B Stat. Methodol.*, vol. 67, no. 1, pp. 91–108, Feb. 2005.

[50] H. Zou, "The adaptive lasso and its oracle properties," *J. Amer. Stat. Assoc.*, vol. 101, no. 476, pp. 1418–1429, Dec. 2006.

[51] T. Park and G. Casella, "The Bayesian Lasso," *J. Amer. Stat. Assoc.*, vol. 103, no. 482, pp. 681–686, 2008.

[52] C. Leng, M.-N. Tran, and D. Nott, "Bayesian adaptive lasso," *Ann. Inst. Stat. Math.*, vol. 66, no. 2, pp. 221–244, Sep. 2013.

[53] K. Shimamura, M. Ueki, S. Kawano, and S. Konishi, "Bayesian generalized fused lasso modeling via NEG distribution," *Commun. Statist.—Theory Methods*, vol. 48, no. 16, pp. 4132–4153, Nov. 2018, doi: 10.1080/03610926.2018.1489056.

[54] M. West, "On scale mixtures of normal distributions," *Biometrika*, vol. 74, no. 3, pp. 646–648, 1987.

[55] J. Griffin and P. Brown, "Alternative prior distributions for variable selection with very many more variables than observations," Univ. Warwick, Coventry, U.K., Tech. Rep. 05-10, May 2005.

[56] E. L. Lehmann and G. Casella, *Theory Point Estimation.* 2nd ed. New York, NY, USA: Springer, 1998.

[57] M. Kyung, J. Gill, M. Ghosh, and G. Casella, "Penalized regression, standard errors, and Bayesian lassos," *Bayesian Anal.*, vol. 5, no. 2, pp. 369–411, Jun. 2010.

[58] A. Gelman and J. Carlin, *Bayesian Data Analysis*, 3rd ed., London, U.K.: Chapman & Hall, 2003.

[59] M. Kalisch and P. Buhlmann, "Estimating high-dimensional directed acyclic graphs with the pcalgorithm," *J. Mach. Learn. Res.*, vol. 8, pp. 613–636, May 2007.

[60] J. E. Aten, T. F. Fuller, A. J. Lusis, and S. Horvath, "Using genetic markers to orient the edges in quantitative trait networks: The NEO software," *BMC Syst. Biol.*, vol. 2, no. 1, p. 34, 2008.

YAN LI received the B.S. degree from the College of Software Engineering, Jilin University, in 2013, where she is currently pursuing the Ph.D. degree with the College of Computer Science and Technology. From 2016 to 2017, she was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Miami. Her research interests include machine learning, big data analysis with applications in bioinformatics and computational biology.

DAYOU LIU received the B.S. degree from the Department of Physics, Jilin University, China, in 1966, and the M.S. degree from the College of Computer Science and Technology, Jilin University, in 1982.

Since 1990, he has been a Professor with the College of Computer Science and Technology, Jilin University. From 1993 to 2004, he was the Dean of the College of Computer Science and Technology, Jilin University, and the Dean of the Faculty of Informatics, from 2006 to 2011. He has authored eight books and published more than 300 articles which have more than 4000 citations by other colleagues. His research interests include artificial intelligence, knowledge engineering, statistical learning, data mining, and expert systems. He is the Distinguished Member of the China Computer Federation (CCF), the Honorary Chair of the Computer Federation of the Jilin Province of China, and the Honorary Committee Member of the CCF Technical Committee on Artificial Intelligence and Pattern Recognition. He received the State Scientific and Technological Progress Award, China, in 2006.

JIANFENG CHU received the M.S. and Ph.D. degrees from the College of Computer Science and Technology, Jilin University, Changchun. He is currently an Associate Professor with the College of Computer Science and Technology, Jilin University. His research interests include data security and privacy, big data analysis, and machine learning.

YUNGANG ZHU received the Ph.D. degree in computer science from Jilin University, China, in 2012. He was a Visiting Research Fellow or a Postdoctoral Fellow with the Vienna University of Technology, Austria, the Dresden University of Technology, Germany, and the University of Trento, Italy. He is currently an Assistant Professor with the College of Computer Science and Technology, Jilin University. He has authored or coauthored over ten articles on international journals or conferences. His current research interests include probabilistic graphical models, information fusion, statistical machine learning, and data mining, with applications to knowledge engineering.

Dr. Zhu is the Committee Member of the CCF Computer Applications Technical Committee and the CAAI Intelligent Service Technical Committee. He served in the program committee for several IEEE international conferences. He serves as an Associate Editor for the IEEE *Canadian Journal of Electrical and Computer Engineering*.

JIE LIU received the Ph.D. degree in computer science from Jilin University, China, in 2007. She is currently an Associate Professor with the College of Computer Science and Technology. Her research interests include data mining and pattern recognition.

XIAOCHUN CHENG received the B.Eng. degree in computer software engineering and the Ph.D. degree in computer science from Jilin University, in 1992 and 1996, respectively. Since 2012, he has been a Computer Science EU Project Coordinator with Middlesex University. He is a member of the IEEE SMC Technical Committee on Enterprise Information Systems, the IEEE SMC Technical Committee on Computational Intelligence, the IEEE SMC Technical Committee on Cognitive Computing, the IEEE SMC Technical Committee on Intelligent Internet Systems, the IEEE Communications Society Communications and Information Security Technical Committee, the BCS Information Security Specialist Group, the BCS Cybercrime Forensics Specialist Group, and the BCS Artificial Intelligence Specialist Group.