Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations

Xiaodong Cai[1,*], Juan Andrés Bazerque[2], Georgios B. Giannakis[2]
1 Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33146, USA
2 Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA
∗ E-mail: x.cai@miami.edu

Abstract

Integrating genetic perturbations with gene expression data not only improves accuracy of regulatory network topology inference, but also enables learning of causal regulatory relations between genes. Although a number of methods have been developed to integrate both types of data, the desiderata of efficient and powerful algorithms still remains. In this paper, sparse structural equation models (SEMs) are employed to integrate both gene expression data and *cis*-expression quantitative trait loci (*cis*-eQTL), for modeling gene regulatory networks in accordance with biological evidence about genes regulating or being regulated by a small number of genes. A systematic inference method named sparsity-aware maximum likelihood (SML) is developed for SEM estimation. Using simulated directed acyclic or cyclic networks, the SML performance is compared with that of two state-of-the-art algorithms: the adaptive Lasso (AL) based scheme, and the QTL-directed dependency graph (QDG) method. Computer simulations demonstrate that the novel SML algorithm offers significantly better performance than the AL-based and QDG algorithms across all sample sizes from 100 to 1,000, in terms of detection power and false discovery rate, in all the cases tested that include acyclic or cyclic networks of 10, 30 and 300 genes. The SML method is further applied to infer a network of 39 human genes that are related to the immune function and are chosen to have a reliable eQTL per gene. The resulting network consists of 9 genes and 13 edges. Most of the edges represent interactions reasonably expected from experimental evidence, while the remaining may just indicate the emergence of new interactions. The sparse SEM and efficient SML algorithm provide an effective means of exploiting both gene expression and perturbation data to infer gene regulatory networks. An open-source computer program implementing the SML algorithm is freely available upon request.

Author Summary

Deciphering the structure of gene regulatory networks is crucial for understanding gene functions and cellular dynamics, as well as system-level modeling of individual genes and cellular functions. Computational methods exploiting gene expression and other types of data generated from high-throughput experiments provide an efficient and low-cost means of inferring gene networks. Sparse structural equation models are employed to: i) integrate both gene expression and genetic perturbation data for inference of gene networks; and, ii) develop an efficient sparsity-aware inference algorithm. Computer simulations corroborate that the novel algorithm markedly outperforms state-of-the-art alternatives. The algorithm is further applied to infer a real human gene network unveiling possible interactions between several genes. Since gene networks can be perturbed not only by genetic variations but also by other means such as gene copy number changes, gene knockdown or controlled gene over-expression, this paper's method can be applied to a number of practical scenarios.

Introduction

Genes in living organisms do not function in isolation, but may interact with each other and act together forming intricate networks [1]. Deciphering the structure of gene regulatory networks is crucial for understanding gene functions and cellular dynamics, as well as for system-level modeling of individual genes and cellular functions. Although physical interactions among individual genes can be experimentally deduced (e.g., by identifying transcription factors and their regulatory target genes or discovering protein-protein interactions), such experimental approach is time-consuming and labor intensive. Given the explosive number of combinations of genes involved in any possible gene interaction, such an approach may not be practically feasible to reconstruct or "reverse engineer" gene networks. On the other hand, technological advances allow for high-throughput measurement of gene expression levels to be carried out efficiently and in a cost-effective manner. These genome-wide expression data reflect the state of the underlying network in a specific condition and provide valuable information that can be fruitfully exploited to infer the network structure.

Indeed, a number of computational methods have been developed to infer gene networks from gene expression data. One class leverages a similarity measure, such as the correlation or mutual information present in pairs of genes, to construct a so-termed co-expression or relevance network [2, 3]. Another approach relies on Gaussian graphical models with edges being present (absent) if the corresponding gene pairs are conditionally dependent (respectively independent), given expression levels of all other genes [4, 5]. While the approach based on Gaussian graphical models entails undirected graphs, directed acyclic graphs (DAGs) or Bayesian networks have also been employed to infer the dependency structure among genes [6, 7]. The fourth approach employs linear regression models and associated inference methods to find the dependency among genes and to infer gene networks [8–11]. Finally, while these approaches use gene expression data in the steady-state, several methods exploiting time-series expression data have also been reported; see e.g., [12, 13] and references therein.

Recently, gene expression data from gene-knockout experiments have been combined with time series comprising gene expression data with perturbations to considerably improve the accuracy of network inference [14]. When a gene is knocked out or silenced, expression levels of other genes are perturbed. Different from using gene expression levels of the original network alone, comparing gene expression levels in the perturbed network with those in the original network reveals extra information about the underlying network structure. Gene perturbations can be performed with other experimental approaches such as controlled gene over-expression and treatment of cells with certain chemical compounds [8, 9]. However, these gene perturbation experiments may not be feasible for all genes or organisms. To overcome this hurdle, one can exploit naturally occurring genetic variations that can be viewed as perturbations to gene networks [15]. More importantly, such genetic variations enable inference of the causal relationship between different genes or between genes and certain phenotypes.

Several approaches are available to capitalize on both genetic variations and gene expression data for inference of gene networks. The first approach models a gene network as a Bayesian network, and then infers the network by incorporating prior information about the network obtained from expression quantitative trait loci (eQTLs) [16–18]. In the second approach, a likelihood test is employed to search for a casual model that "best" explains the observed gene expression and eQTL data [19–23]. The third approach relies on the structural equation model (SEM) to infer gene [24–27] or phenotype networks [28–34]. While these approaches focus on inference of gene networks incorporating information from eQTL, another approach employs both phenotype and QTL genotype data to jointly decipher the phenotype network and identify eQTLs that are causal for each phenotype [35]. Logsdon and Mezey [26] proposed an adaptive Lasso (AL) [36] based algorithm to infer gene networks modeled with an SEM. They compared the performance of a number of methods using simulated directed acyclic or cyclic networks. Their simulations showed that the AL-based algorithm outperformed all other methods tested. Despite its superiority over other methods, the AL-based algorithm does not fully exploit the structure of the SEM. Therefore, it is expected that a more systematic inference algorithm may significantly improve the

performance of the SEM-based approach.

Motivated by the fact that gene networks or more general biochemical networks are sparse [8,37–39], a sparse SEM is advocated in this paper to infer gene networks from both gene expression and eQTL data. Incorporating network sparsity constraints, a sparsity-aware maximum likelihood (SML) algorithm is developed for network topology inference. The core technique used is to maximize the likelihood function regularized by the ℓ_1-norm of the parameter vector determining the network structure. The ℓ_1-norm controls complexity of the SEM, and thus yields a sparse network. The key innovative element of the SML algorithm is a block coordinate ascent method derived to maximize the ℓ_1-regularized likelihood function, which makes the SML algorithm computationally efficient. The simulations provided demonstrate that the novel SML algorithm offers significantly better performance than the two state-of-the-art algorithms: the AL [26], and the QDG algorithm [21]. The SML algorithm is further applied to infer a human network of 39 human genes related to the immune function.

Results

Sparse SEM model for gene regulatory networks

Consider expression levels of N_g genes from N individuals measured using e.g., microarray or RNA-seq. Let $\mathbf{y}_i := [y_{i1}, \ldots, y_{iN_g}]^T$ denote the $N_g \times 1$ vector collecting the expression levels of these N_g genes of individual i. Suppose that a set of perturbations to these genes has been also observed. These perturbations can be due to naturally occurring genetic variations near or within the genes, gene copy number changes, gene knockdown by RNAi or controlled gene over-expression. In this paper, focus is placed on genetic variations observed at eQTLs, although the network model and the inference method described in the next section are also applicable to cases where other perturbations are available. As in [26], it is assumed that each gene has at least one *cis*-eQTL so that the structure of the underlying gene network is uniquely identifiable. Let $\mathbf{x}_i := [x_{i1}, \ldots, x_{iN_q}]^T$ denote the genotype of $N_q \geq N_g$ eQTLs of individual i. The goal is to infer the network structure of the N_g genes from the available gene expression measurements $\mathbf{y}_i$, $i = 1, \ldots, N$, and eQTL observations $\mathbf{x}_i$, $i = 1, \ldots, N$.

As in [25,26], the gene network is postulated to obey the SEM

$$\mathbf{y}_i = \mathbf{B}\mathbf{y}_i + \mathbf{F}\mathbf{x}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, N \tag{1}$$

where $N_g \times N_g$ matrix $\mathbf{B}$ contains unknown parameters defining the network structure; $N_g \times N_q$ matrix $\mathbf{F}$ captures the effect of each eQTL; $N_g \times 1$ vector $\boldsymbol{\mu}$ accounts for possible model bias; and $N_g \times 1$ vector $\boldsymbol{\epsilon}_i$ captures the residual error, which is modeled as a zero-mean Gaussian vector with covariance $\sigma^2\mathbf{I}$, where $\mathbf{I}$ denotes the $N_g \times N_g$ identity matrix. It is assumed that no self-loops are present per gene, which implies that the diagonal entries of $\mathbf{B}$ are zero. As mentioned in [26], lack of self-loops and a diagonal covariance matrix of $\boldsymbol{\epsilon}_i$ are commonly assumed in almost all graph-based network inference methods. It is further assumed that the loci of N_q eQTLs have been determined using an existing eQTL method, but the effective size of each eQTL is unknown. Therefore, $\mathbf{F}$ has N_q unknown entries whose locations are known and $N_gN_q - N_q$ remaining zero entries (for instance $\mathbf{F}$ is a diagonal matrix when $N_q = N_g$).

The network inference task is to estimate $N_g(N_g - 1)$ unknown entries of $\mathbf{B}$, and as a byproduct, the N_q unknown entries of $\mathbf{F}$. Without any knowledge about the network, no restriction is imposed on the structure specified by $\mathbf{B}$. Therefore, the network is considered as a general directed graph that can possibly be a directed cyclic graph (DCG) or a DAG. Network inference is challenging since the number of unknowns to be estimated is very large for a moderately large N_g. Note that under the assumption that each gene has at least one *cis*-eQTL, the "Recovery" Theorem in [26] guarantees that the network is identifiable for both DCGs and DAGs.

As discussed in [8,37–39], gene regulatory networks 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. Taking into account sparsity, only a relatively small number of the entries of $\mathbf{B}$ are nonzero. These nonzero entries determine the network structure and the regulatory effect of one gene on other genes. The SEM in (1) under the aforementioned sparsity assumption will be henceforth referred to as the sparse SEM. Exploiting the sparsity inherent to the network, an efficient and powerful algorithm for network inference will be developed in the ensuing section.

Sparsity-aware inference method

Upon defining $\mathbf{Y} := [\mathbf{y}_1, \ldots, \mathbf{y}_N]$, $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, and $\mathbf{E} := [\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_N]$, the SEM in (1) can be compactly written as $\mathbf{Y} = \mathbf{B}\mathbf{Y} + \mathbf{F}\mathbf{X} + \boldsymbol{\mu}\mathbf{1}^T + \mathbf{E}$, where $\mathbf{1}$ is the $N \times 1$ vector of all-ones. Given $\mathbf{X}$ and $\mathbf{Y}$, the log-likelihood function can be written as

$$\begin{aligned}\log p(\mathbf{Y}|\mathbf{X}; \mathbf{B}, \mathbf{F}, \boldsymbol{\mu}) =& \frac{N}{2} \log |\det(\mathbf{I} - \mathbf{B})|^2 - \frac{N N_g}{2} \log(2\pi\sigma^2) \\ & - \frac{1}{2\sigma^2} \|\mathbf{Y} - \mathbf{B}\mathbf{Y} - \mathbf{F}\mathbf{X} - \boldsymbol{\mu}_\epsilon \mathbf{1}^T\|_F^2 \end{aligned} \tag{2}$$

where $\det(\cdot)$ denotes matrix determinant, and $\|\cdot\|_F$ denotes the Frobenius norm.

As mentioned earlier, $\mathbf{B}$ is a sparse matrix having most entries equal to zero. In order to obtain a sparse estimate of $\mathbf{B}$, the natural approach is to maximize the log likelihood regularized by the weighed ℓ_1-norm term $\|\mathbf{B}\|_{1,W} := \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} w_{ij}|B_{ij}|$, where B_{ij} denotes the (i,j)th entry of $\mathbf{B}$. In a linear regression model, it is well known that the ℓ_1-regularized least-squares estimation also known as Lasso [40] can yield a sparse estimate of the regression coefficient vector. Similarly, the ℓ_1-regularized maximum likelihood (ML) approach used here is expected to shrink most of the entries of $\mathbf{B}$ toward zero, thereby yielding a sparse matrix. It is easy to show that maximizing $\log p(\mathbf{Y}|\mathbf{X}; \mathbf{B}, \mathbf{F}, \boldsymbol{\mu})$ with respect to (w.r.t.) $\boldsymbol{\mu}$ yields $\hat{\boldsymbol{\mu}} = (\mathbf{I} - \mathbf{B})\bar{\mathbf{y}} - \mathbf{F}\bar{\mathbf{x}}$, where $\bar{\mathbf{y}} = \sum_{n=1}^N \mathbf{y}_n/N$ and $\bar{\mathbf{x}} = \sum_{n=1}^N \mathbf{x}_n/N$. Upon defining $\tilde{\mathbf{y}}_n := \mathbf{y}_n - \bar{\mathbf{y}}$, $\tilde{\mathbf{x}}_n := \mathbf{y}_n - \bar{\mathbf{x}}$, $\tilde{\mathbf{Y}} := [\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_N]$, $\tilde{\mathbf{X}} := [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_N]$, and substituting $\hat{\boldsymbol{\mu}}$ for $\boldsymbol{\mu}$ in (2), the proposed ℓ_1-penalized ML estimation approach yields

$$\begin{aligned}(\hat{\mathbf{B}}, \hat{\mathbf{F}}) = \arg\max_{\mathbf{B},\mathbf{F}} & N\sigma^2 \log |\det(\mathbf{I} - \mathbf{B})| - \frac{1}{2}\|\tilde{\mathbf{Y}} - \mathbf{B}\tilde{\mathbf{Y}} - \mathbf{F}\tilde{\mathbf{X}}\|_F^2 - \lambda\|\mathbf{B}\|_{1,W} \\ & \text{subject to } \; B_{ii} = 0, \forall i = 1, \ldots, N_g, \; F_{jk} = 0, \; \forall (j,k) \in \mathcal{S}_q \end{aligned} \tag{3}$$

where $\mathcal{S}_q$ denotes the set of row and column indices of the entries of $\mathbf{F}$ known to be zero. As assumed earlier, each phenotype has at least one *cis*-eQTL that has been identified, which implies that the locations of nonzero entries of $\mathbf{F}$ or equivalently the set $\mathcal{S}_q$ is known. However, our sparse SEM and inference method are also applicable to more general cases where some or all phenotypes have *cis*-eQTLs that have not been identified. In these cases, the locations of nonzero entries of $\mathbf{F}$ corresponding to the unidentified *cis*-eQTLs are unknown. We can form a weighted ℓ_1-norm of the entries of $\mathbf{F}$ excluding those corresponding to the identified *cis*-eQTL and then add a penalty term involving this ℓ_1-norm to the objective function in (3). This new optimization problem can be solved efficiently using a method modified from the one solving (3), as it is described in the supporting text S1.

Weights w_{ij} in the penalty term are introduced to improve estimation accuracy in line with the AL [36]. They are selected as $1/\tilde{B}_{ij}$, where $\tilde{B}_{ij}$ is found using a preliminary estimate of $\mathbf{B}$ obtained via ridge regression as

$$\begin{aligned}(\tilde{\mathbf{B}}, \tilde{\mathbf{F}}) = \arg\min_{\mathbf{B},\mathbf{F}} & \frac{1}{2}\|\tilde{\mathbf{Y}} - \mathbf{B}\tilde{\mathbf{Y}} - \mathbf{F}\tilde{\mathbf{X}}\|_F^2 + \rho\|\mathbf{B}\|_F^2 \\ & \text{subject to } \; B_{ii} = 0, \forall i = 1, \ldots, N_g, \; F_{jk} = 0, \; \forall (j,k) \in \mathcal{S}_q. \end{aligned} \tag{4}$$

The sparsity-controlling parameters λ in (3) and ρ in (4) are selected via cross validation (CV), while σ^2 is estimated as the sample variance of the error using $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{F}}$. In adaptive Lasso based linear

regression [36], Zou suggested using the ordinary least squares (OLS) estimate to determine the weights; if the OLS estimate does not exist due to, e.g., collinearity, Zou suggested the estimate obtained from ridge regression, although it remains to show if the ridge regression estimate is consistent in this case and if the resulting adaptive Lasso yields the desired oracle properties. If OLS is used for estimating $\mathbf{B}$ and $\mathbf{F}$ in the SEM, the solution usually does not exist since the number of unknowns is typically larger than the number of samples. However, even in this case the solution can always be obtained from ridge regression as in (4). Moreover, every entry of the solution is typically nonzero, which yields a finite weight for every variable, and thus every variable will be included in the following ℓ_1-penalized ML procedure. An alternative approach is to replace the weighed ℓ_1-norm in (3) with an unweighted ℓ_1-norm to obtain a preliminary estimate of $\mathbf{B}$ and then calculate the weights from this preliminary estimate, as in [26]. However, the unweighted ℓ_1-penalized ML procedure may shrink many variables to zero and exclude them from the weighted ℓ_1-penalized ML estimator, possibly yielding a biased estimate. For this reason, the inference method in this paper uses ridge regression to determine $\{w_{ij}\}$, with the additional advantage of (4) admitting a closed-form solution.

A block diagram of the novel inference algorithm, abbreviated as the sparsity-aware maximum likelihood (SML) algorithm, is depicted in Figure 1. The first and third blocks in Figure 1 perform cross-validation to select optimal parameters ρ and λ to be used in (3) and (4), respectively (see the description of the cross-validation procedure in the Supporting text S1.) The third block produces weights $\{w_{ij}\}$ and error-variance estimate $\hat{\sigma}_e^2$ after solving (4). Finally, the fourth block takes data $\mathbf{X}$ and $\mathbf{Y}$ together with λ, $\{w_{ij}\}$ and $\hat{\sigma}_e^2$ and solves (3) to yield $\hat{\mathbf{B}}$, representing the SML estimator for $\mathbf{B}$ in (1) and revealing the genetic-interaction network. As it will be described in the Methods section, (4) is separable across rows of $\mathbf{B}$ and $\mathbf{F}$, and each row of $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{F}}$ becomes available in closed form [cf. (8)-(9)]. The ℓ_1-regularized ML problem (3) is solved efficiently using a novel block coordinate ascent iterative scheme given by (11)-(16) in the Methods section. Precise description of the overall SML algorithm is also presented in the Methods section as Algorithm 1, which was used to yield an executable computer program.

Simulation studies and performance comparison of inference algorithms

In their simulation studies, Logsdon and Mezey [26] compared the performance of their AL-based algorithm with that of several other algorithms including the PC-algorithm [41,42], the QDG algorithm [21], the QTLnet algorithm [35], and the NEO algorithm [22]. In two out of four simulation setups, the AL outperformed all other algorithms; and in the other two simulation setups, the AL and QDG algorithms exhibited comparable performance, but consistently outperformed the other two algorithms. Logsdon and Mezey [26] also considered other existing algorithms [25,43], but these were deemed either computationally too demanding [43] or prohibitively complex [25]. For these reasons, the AL and QDG algorithms are regarded as state-of-the-art in the field. Their performance was compared against this paper's SML algorithm.

Following the setup of Logsdon and Mezey [26], two types of acyclic gene networks were simulated first: one with 10 genes and another with 30 genes. Specifically, a random DAG of 10 or 30 nodes with an expected $N_e = 3$ edges per node was generated by creating directed edges between two randomly picked nodes. Care was taken to avoid any cycle in the simulated graph. If an edge from node j to node i was emerging, B_{ij} was generated from a random variable uniformly distributed over the interval $(0.5, 1)$ or $(-1, -0.5)$; otherwise, $B_{ij} = 0$. The genotype per eQTL was simulated from an F2 cross. Values 1 and 3 were assigned to two homozygous genotypes, respectively, and 2 to the heterozygous genotype. Hence, X_{ij} was generated as a ternary random variable taking values $\{1, 3, 2\}$ with corresponding probabilities $\{0.25, 0.25, 0.5\}$. Matrix $\mathbf{F}$ was the $N_g \times N_g$ identity matrix, E_{ij} was sampled from a Gaussian distribution with zero mean and variance 10^{-2}, and $\boldsymbol{\mu}$ was set to zero. Finally, $\mathbf{Y}$ was calculated from $\mathbf{Y} = (\mathbf{I} - \mathbf{B})^{-1}(\mathbf{F}\mathbf{X} + \mathbf{E})$.

For each type of gene network, 100 realizations or replicates of the network were generated, and then the SML, the AL and the QDG algorithms were run to infer the network topology. When running the

SML algorithm, 10-fold CV was employed to determine the optimal values of parameters λ and ρ and then use these values to infer the network. An edge from gene j to i was deemed present if $\hat{B}_{ij} \neq 0$. The AL algorithm also automatically ran using CV to determine the values of its parameters. For 100 replicates of the network, N_t counted the total number of edges, $\hat{N}_t$ denoted the total number of edges detected by the inference algorithm. Among $\hat{N}_t$ detected edges, N_{true} stands for the number of true edges presented in the simulated networks, and N_{false} for the number of false edges. The power of detection (PD) was then found as N_{true}/N_t, and the false discovery rate (FDR) as $N_{false}/\hat{N}_t$. The PD and the FDR of the SML, AL, and QDG algorithms for different sample sizes are depicted in Figure 2. It is seen from Figures 2(a) and (c) that the PD of the SML algorithm exceeds 0.9 for both networks across all sample sizes, whereas the PD of the AL algorithm is about 0.65 for $N_g = 10$ and 0.35 for $N_g = 30$. The PD of the QDG algorithm is even lower ranging from 0.22 to 0.33. As shown in Figures 2(b) and (d) , the FDR of the SML algorithm is on the order of 10^{-3} for most sample sizes, and is much lower than that of the AL and QDG algorithms, which is about 0.3 for $N_g = 10$ and over the range from 0.31 to 0.6 for $N_g = 30$.

Two types of *cyclic* networks were subsequently simulated: one with 10 genes and the other with 30 genes. The average number of edges per gene is again equal to 3. The same procedures used in simulating acyclic networks described earlier were employed, except that DCGs instead of DAGs were simulated. Again, 100 replicates for each type of the networks were randomly generated. The PD and the FDR of three algorithms are depicted in Figure 3. As shown in Figure 3(a) and (c) , the PD of the SML algorithm is between 0.83 and 0.9, whereas the PD of the AL algorithm is about 0.52 for $N_g = 10$ and 0.29 for $N_g = 30$, and the PD of the QDG algorithm is between 0.16 and 0.28. As shown in Figures 3(b) and (d) , the FDR of the SML algorithm is < 0.01, which is much smaller than that of the AL and QDG algorithms over the range from 0.33 to 0.68. For the convenience of comparison, the results in Figures 2 and 3 at sample size 500 are summarized in Table 1.

As confirmed by Figures 2 and 3, the SML algorithm offers much better performance in terms of PD and FDR than the AL and QDG algorithms. However, these results were obtained for gene networks of small size. To test performance of the SML algorithm for networks of relatively large size, an acyclic network of 300 genes was simulated with an expected $N_e = 1$ edge per node, and randomly generated 10 replicates of the network. PD and FDR of the SML and AL algorithms obtained from these replicates are depicted in Figure 4. The PD of SML exceeds 0.99 across all sample sizes from 100 to 1,000, whereas that of the AL algorithm is about 0.04 for sample sizes from 100 to 500, and gradually increases to 0.42 at the sample size of 1,000. The FDR of SML stays below 10^{-4} for sample sizes from 400 to 1,000, whereas the FDR of the AL algorithm is on the order of 10^{-2} for the same sample size. When the sample size is relatively small (in the range from 100 to 300), the FDR of SML is higher than that of the AL algorithm, but it is still relatively small (< 0.2). Note that the AL algorithm essentially does not work for sample sizes $N \leq 500$, since its power is too small. All simulation results show that the novel SML algorithm significantly outperforms the AL and QDG algorithms in terms of PD and FDR.

An extra set of simulations assessing the stability of SML is described in the section of "Stability of model selection under CV perturbations" in supporting text S1. As an alternative to CV, stability selection (STS) [44] provides a means of selecting an appropriate sparsity level to guarantee that the FDR is less than a theoretical upper bound. The STS procedure was applied to the SML algorithm as described in the supporting text S1, and was used with the selection probability cutoff $\delta = 0.8$ and an upper bound or target FDR=0.1 in simulations for the networks in Figures 2[(c) and (d)] and 3 [(c) and (d)]. As shown in Figure **??**, the FDR of the STS is indeed much smaller than the target FDR and almost uniform across different sample sizes, but the PD of the STS is smaller than that of CV. In fact, the FDR of the STS is on the same order as that of the CV except at the sample size of 100 for the DAG. As seen from these simulation results, although the STS guarantees a FDR upper bound, this upper bound is loose for the simulation setups tested, which may sacrifice detection power. Nevertheless, the STS procedure can select a set of stable variables as described in [44] and verified by our simulations.

So far, all the simulated data were generated with noise variance $\sigma^2 = 0.01$. Next, the performance of SML was analyzed for simulated networks of 30 genes, when σ^2 was increased to 0.05 and N_e was changed from 3 to 1 or 5. Reducing N_e from 3 to 1 improved the performance of SML for most of the sample sizes, as it is depicted in Figure 5, withstanding the increase in the noise variance. Increasing N_e at constant σ^2, or increasing σ^2 at constant N_e degraded the performance, most notably in the later case. Comparing Figure 5 with Figures 2 and 3 [(c) and (d)] demonstrates that in both cases the SML estimates still achieve higher detection power and lower FDR than those estimates obtained with the AL algorithm for $N_e = 3$ and $\sigma^2 = 0.01$.

Inference of a network of immune-related human genes

Pickrell *et al.* [45] used RNA-Seq technology to sequence RNA from 69 lymphoblastoid cell lines derived from unrelated Nigerian individuals extensively genotyped by the International HapMap Project [46]. For each gene, they evaluated possible associations between its gene expression level calculated from RNA-Seq reads and all 3.8 million single nucleotide polymorphisms (SNPs) using the genotypes from phases II and III of the HapMap Project. At FDR=0.1, they identified 929 genes or putative new exons that have eQTLs within 200kb of the gene or the exon. From these 929 genes, 39 genes that are related to immune functions were selected manually by an expert as mentioned in the Acknowledgements section; expression levels and the genotypes of the eQTLs of these 39 genes in 69 individuals were used to infer the underlying regulatory network.

Pickrell *et al.* normalized expression values using quantile normalization before performing eQTL mapping. They also provided a data set that contains the number of reads mapped to each of 929 genes. This data set was obtained and the number of reads for each of 39 genes was normalized with the length of the gene to yield expression value. Such kind of values may better reflect the real expression values than the values normalized with quantile normalization, and thus they were used to infer the network. To ensure the quality of the data, the SAS ROBUSTREG procedure was applied to 69 expression values of each of 39 genes to detect outliers. The default M estimation method of the ROBUSTREG procedure was employed and the outliers were detected at a significance level of 0.05. Several outliers with values much larger than the remaining values were identified and were replaced with the largest non-outlier since it is closest to the outliers. More sophisticated means of revealing and imputing outliers are possible using robust statistical schemes; see e.g., [47]. The genotypes of the eQTLs of the 39 genes were downloaded from HapMap database using the SNP IDs for the eQTL provided by Pickrell *et al.*. About 12% genotypes are missing. These missing genotypes were imputed using the program IMPUTE2 [48]. The name and a brief description of each gene were obtained from DAVID [49] using the Ensembl gene IDs provided by Pickrell *et al.* Information of these 39 genes including their Ensembl gene IDs and names, a brief description of each gene, and HapMap SNP IDs of the associated eQTLs can be found in Table S1 in the supporting information.

The SML algorithm was run with the expression levels and genotypes of eQTLs of these 39 genes. An edge from gene j to i was detected if $\hat{B}_{ij} \neq 0$. To improve the reliability of the detected edges, the SML algorithm was run with stability selection at an FDR ≤ 0.1 using 100 random subsamples, yielding 13 directional edges as shown in Figure 6. The frequency of each edge detected in 100 runs is given in Table **??**. It is interesting to see from Figure 6 that only 9 genes are involved in the network, and the remaining 30 genes are not connected with any other genes and thus not shown in the figure. AL and QDG algorithms were also run with stability selection at an FDR ≤ 0.1 using 100 random subsamples. The edges detected by AL and QDG algorithms and their frequencies are included in Table **??**. The AL algorithm detected only one edge that was not detected by the SML algorithm. The QDG yielded 3 edges, one of which was also detected by the SML algorithm. Comparing the results of three algorithms shows that our SML algorithm detected more edges than the other two algorithms at the same FDR due to its higher detection power as confirmed also by the simulations. When the FDR was increased to ≤ 0.3, the SML algorithm with stability selection yielded a network of 16 genes that have 42 edges

as shown in Figure **??** in the supporting information. Since only 39 genes were used to construct the network, an edge between two genes may not necessarily imply a direct regulatory effect, but may reflect the fact that two genes are either directly linked or very close to each other in the real network that consists of all genes. Particularly, if two genes are co-regulated by another gene which is not included in the 39 genes, these two genes may have a unidirectional or bidirectional edge.

Most edges in Figure 6 are between major histocompatibility complex (MHC) genes (HLA-A, HLA-DPA1, HLA-DQA2, HLA-DQB1, HLA-DRB4 and HLA-DRB5), which is expected since these genes may interact with each other and/or be co-regulated. FCRLA is a member of Fc receptor-like family of genes. It is expressed in B cells and interacts with IgG and IgM [50, 51]. IGH, encoding the heavy chain of immunoglobulin, characterizes the B-cell origin of the samples. Hence, it is not surprising to see an edge between FCRLA and IGH. Interleukin-4-induced gene 1 (IL4I1) was first described in the mouse [52] and subsequently characterized in human B cells [53]. Human IL4I1 is expressed by antigen-presenting cells [54], which may allude to the edge between HLA-A and IL4I1, but this may be speculative since there is no edges between IL4I1 and MHC class II genes in the network. The edges between IGH and HLA-A and between IGH and HLA-DRB4 may reflect the coordinated effect of antibody and MHC as a response to antigens. In fact, IGH is connected to most of MCH genes in Figure **??**, which may imply the wide coordination between the two classes of molecules.

Discussion

Integrating genetic perturbations with gene expression data for inference of gene networks not only improves inference accuracy, but also enables learning of causal regulatory relations among genes. Although much progress has been made recently on the development of inference methods that integrate both types of data, a truly efficient algorithm is missing. The SEM provides a systematic framework to integrate both types of data, and offers flexibility to model both directed cyclic as well as acyclic graphs. However, there is no systematically designed inference method for SEMs of relatively high dimension, which is particularly true for gene networks typically including hundreds or thousands of genes. Traditionally, inference for SEMs has relied on the ML or generalized least-squares methods implemented with a numerical optimization algorithm [55, 56]; but recently, Bayesian alternatives [57] have emerged too, based on Markov chain Monte Carlo simulations [58,59]. These methods not only are computationally intensive, but also may be inaccurate for sparse SEMs of relatively high dimension, since they do not account for sparsity present in the model.

In the context of QTL mapping, Newton's method is employed in [27] to implement the ML method, while the genetic algorithm [60,61] is used in [24,25] to maximize the likelihood function, and in conjunction with a model selection method using a χ^2 test or Occam's window to search for the best network topology. These methods are not scalable to SEMs of relatively high dimension. The AL-based algorithm proposed in [26] is more efficient because it automatically incorporates model selection into the inference process, and also takes into account the sparsity present in gene networks. However, the AL-based scheme borrows the adaptive Lasso [36] optimally designed for the linear regression model instead of the SEM. In contrast, the SML algorithm proposed in this paper directly maximizes the ℓ_1-regularized likelihood function of the SEM, which fully exploits the information present in the data and therefore improves inference accuracy. Moreover, the novel block coordinate ascent method combined with discarding rules can efficiently maximize the ℓ_1-regularized likelihood function, rendering the SML algorithm applicable to SEMs of high dimension. However, unlike the AL-based algorithm, the SML algorithm maximizes a non-convex objective function as given in (3). Although the "Recovery" Theorem in [26] guarantees the identifiability of the network, the algorithm can converge to a local maximum that may not necessarily be coincident with the global maximum corresponding to the optimal network. A common technique for alleviating this problem is to use multiple random initial values. We tested multiple initial values in our simulations and observed that the algorithm converged to the same solution. In Algorithm 1, we used

the pathwise coordinate optimization strategy as used in [62], where the solution of (3) obtained with λ_i was used as the initial point for the run with $\lambda_{i+1} < \lambda_i$. The pertinence of this strategy is corroborated by simulated numerical tests, showing significant performance gains of the SML algorithm in terms of detection power and FDR when compared to the AL-based algorithm.

Comparisons in the Simulation Studies section, as summarized in Figures 2-5, demonstrated that the SML algorithm markedly outperforms two state-of-the-art algorithms: the AL [26] and QDG [21] algorithms. For three directed acyclic networks with number of genes $N_g = 10, 30$ and 300, respectively, the PD of the SML algorithm exceeds 0.9 for all sample sizes from 100 to 1,000, and is greater than 0.99 for most sample sizes. This is much greater than the PD of the AL and QDG algorithm that ranges from 0.004 to 0.67. In fact, The QDG algorithm was too time-consuming to obtain results for $N_g = 300$. The FDR of SML is on the order of 10^{-3} for most sample sizes, which is much smaller than those of the AL and QDG algorithms, that are between 0.25 and 0.6 for $N_g = 10$ and 30. The FDR of the AL algorithm for $N_g = 300$ is between 0.02 and 0.1. The only case where the FDR of SML exceeds that of the AL algorithm is when $N_g = 300$, and the sample size $N < 400$. However, the AL algorithm essentially does not work in this case, since its PD is about 0.04. In the case of directed cyclic networks, all algorithms offer slightly degraded performance when compared to that of directed acyclic networks. However, the SML algorithm still considerably outperforms the AL and QDG algorithms.

Using a limited amount of available data [45], 39 genes related to the immune system and having one eQTL per gene were selected to infer a possible network among these genes. At an FDR $\leq$10% for the detected edges, a network of 9 out of 39 genes containing 13 edges were obtained. An edge between two genes in the inferred network may be an indication of the direct regulator effect, or indirect interaction or co-regulation mediated by some other genes that are not among the 39 genes. The majority of the edges were reasonably expected from the experimental results in the literature, while the remaining edges may represent new interactions to be elucidated.

Structural equation modeling has a long history of about a century, with well-documented contributions to various fields including biology, psychology, econometrics and other social sciences [55,56,63,64]. The model considered in this paper belongs to a class of SEMs with observed variables [55]. The SML algorithm is the first one that is systematically developed for inferring sparse SEMs with observed variables. It is expected to accelerate the application of high-dimensional SEMs not only in biology, but also in other fields.

Methods

Ridge regression

Closed-form solution: Problem (4) can be solved row by row independently in closed form. Let $\mathbf{b}_i^T$, $\tilde{\mathbf{b}}_i^T$, $\mathbf{f}_i^T$, $\tilde{\mathbf{f}}_i^T$ and $\check{\mathbf{y}}_i^T$ denote the ith row of $\mathbf{B}$, $\tilde{\mathbf{B}}$, $\mathbf{F}$, $\tilde{\mathbf{F}}$, and $\tilde{\mathbf{Y}}$, respectively. Then, problem (4) is equivalent to the following problem

$$
\begin{aligned}
&(\tilde{\mathbf{b}}_i, \tilde{\mathbf{f}}_i) = \arg\min_{\mathbf{b}_i, \mathbf{f}_i} \frac{1}{2}\|\check{\mathbf{y}}_i^T - \mathbf{b}_i^T\tilde{\mathbf{Y}} - \mathbf{f}_i^T\tilde{\mathbf{X}}\|_2^2 + \rho\|\mathbf{b}_i\|_2^2 \\
&\text{subject to } \ b_i(i) = 0, \ f_i(k) = 0, \ \forall k \text{ s.t. } (i,k) \in \mathcal{S}_q
\end{aligned} \tag{5}
$$

where $b_i(j)$ stands for the jth element of $\mathbf{b}_i$ and $f_i(k)$ denotes the kth element of $\mathbf{f}_i$.

The constraints in (5) can be imposed directly by discarding elements of $\mathbf{b}_i$ and $\mathbf{f}_i$ known to be zero. To this end, define an $(N_g - 1) \times 1$ vector $\check{\mathbf{b}}_i := [b_i(1), \ldots, b_i(i-1), b_i(i+1) \ldots, b_i(N_g)]^T$ and a vector $\check{\mathbf{f}}_i$ collecting the entries of $\mathbf{f}_i$ whose indexes are not in $\mathcal{S}_q(i) := \{k \in \mathbb{N} : \ (i,k) \in \mathcal{S}_q\}$. Let $\bar{\mathbf{b}}_i$ and $\bar{\mathbf{f}}_i$ denote the solution for $\check{\mathbf{b}}_i$ and $\check{\mathbf{f}}_i$, respectively. Similarly, let $\check{\mathbf{Y}}_i$ be a sub-matrix of $\tilde{\mathbf{Y}}$ formed by removing the ith row of $\tilde{\mathbf{Y}}$, and $\check{\mathbf{X}}_i$ collecting those rows of $\tilde{\mathbf{X}}$ whose indexes are not in $\mathcal{S}_q(i)$. Under these definitions,

(5) is equivalent to

$$(\bar{\mathbf{b}}_i, \bar{\mathbf{f}}_i) = \arg\min_{\check{\mathbf{b}}_i, \check{\mathbf{f}}_i} \frac{1}{2}\|\check{\mathbf{y}}_i - \check{\mathbf{Y}}_i^T \check{\mathbf{b}}_i - \check{\mathbf{X}}_i^T \check{\mathbf{f}}_i\|_2^2 + \rho\|\check{\mathbf{b}}_i\|_2^2. \tag{6}$$

Minimizing for $\check{\mathbf{f}}_i$ first, one arrives at

$$\check{\mathbf{f}}_i = \left(\check{\mathbf{X}}_i \check{\mathbf{X}}_i^T\right)^{-1} \check{\mathbf{X}}_i \left(\check{\mathbf{y}}_i - \check{\mathbf{Y}}_i \check{\mathbf{b}}_i\right). \tag{7}$$

Substituting (7) into (6) after defining $\mathbf{P}_i := \mathbf{I} - \check{\mathbf{X}}_i^T \left(\check{\mathbf{X}}_i \check{\mathbf{X}}_i^T\right)^{-1} \check{\mathbf{X}}_i$, yields

$$\bar{\mathbf{b}}_i = \arg\min_{\check{\mathbf{b}}_i} \frac{1}{2}\|\mathbf{P}_i \check{\mathbf{y}}_i - \mathbf{P}_i \check{\mathbf{Y}}_i^T \check{\mathbf{b}}_i\|_2^2 + \rho\|\check{\mathbf{b}}_i\|_2^2,$$

which is a standard ridge regression problem with solution given by

$$\bar{\mathbf{b}}_i = \left(\check{\mathbf{Y}}_i \mathbf{P}_i \check{\mathbf{Y}}_i^T + \rho\mathbf{I}\right)^{-1} \check{\mathbf{Y}}_i^T \mathbf{P}_i \check{\mathbf{y}}_i. \tag{8}$$

Finally, substituting (8) into (7) yields

$$\bar{\mathbf{f}}_i = \left(\check{\mathbf{X}}_i \check{\mathbf{X}}_i^T\right)^{-1} \check{\mathbf{X}}_i \left(\mathbf{I} - \check{\mathbf{Y}}_i \left(\check{\mathbf{Y}}_i \mathbf{P}_i \check{\mathbf{Y}}_i^T + \rho\mathbf{I}\right)^{-1} \check{\mathbf{Y}}_i^T \mathbf{P}_i\right) \check{\mathbf{y}}_i. \tag{9}$$

Vectors $\tilde{\mathbf{b}}_i$ and $\tilde{\mathbf{f}}_i$ are obtained by inserting zeros into $\bar{\mathbf{b}}_i$ and $\bar{\mathbf{f}}_i$ at appropriate positions specified by the constraints in (5). Collecting $\tilde{\mathbf{b}}_i$ and $\tilde{\mathbf{f}}_i$, $i = 1, \ldots, N_g$, yields the solution of (4), namely $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{F}}$.

Parameter ρ is required to solve (4). A K-fold CV scheme is adopted for this purpose with typical choices of $K = 5$ or 10, as suggested in [65]. A detailed description of the CV procedure [65] is given in supporting text S1.

ℓ_1-regularized ML method

Coordinate-ascent algorithm: Solving (3) is performed by a cyclic block-coordinate ascent iteration. Consider a specific cycle where estimates of $\mathbf{B}$ and $\mathbf{F}$ obtained in the previous cycle are denoted by $\hat{\mathbf{B}}$ and $\hat{\mathbf{F}}$, respectively. The first step of the cycle entails maximizing the objective function in (3) w.r.t. $\mathbf{F}$ with $\mathbf{B}$ fixed to $\hat{\mathbf{B}}$, which yields a new estimate of $\mathbf{F}$ denoted as $\hat{\mathbf{F}}^{\text{new}}$. This step coincides with the minimization of the objective function in (4) w.r.t. $\mathbf{F}$, which admits a closed-form solution per row given by (7). In each of the next $N(N-1)$ steps of the cycle, the objective function in (3) is maximized w.r.t. a single entry of $\mathbf{B}$, namely $B_{ij},\ i \neq j$, with the remaining entries of $\mathbf{B}$ equal to the corresponding entries of $\hat{\mathbf{B}}$ and $\mathbf{F} = \hat{\mathbf{F}}^{\text{new}}$. An expression for the new estimate of B_{ij}, $\hat{B}_{ij}^{\text{new}}$ is derived next.

Define matrix $\hat{\mathbf{B}}(B_{ij}) := \hat{\mathbf{B}} + \mathbf{e}_i \mathbf{e}_j^T (B_{ij} - \hat{B}_{ij})$ having all entries equal to those of $\hat{\mathbf{B}}$ except for its (i,j)th entry, which is replaced by the variable B_{ij}, where $\mathbf{e}_i$ and $\mathbf{e}_j$ denote the ith and jth canonical vectors in $\mathbb{R}^{N_g}$, respectively. Then, the objective in (3) can be written as

$$f_{ij}(B_{ij}) = N\hat{\sigma}^2 \log|\det(\mathbf{I} - \hat{\mathbf{B}}(B_{ij}))| - \frac{1}{2}\|\tilde{\mathbf{Y}} - \hat{\mathbf{B}}(B_{ij})\tilde{\mathbf{Y}} - \hat{\mathbf{F}}^{\text{new}}\tilde{\mathbf{X}}\|_F^2 - \lambda w_{ij}|B_{ij}|. \tag{10}$$

Upon re-arranging and discarding constant terms, (10) simplifies to

$$g_{ij}(B_{ij}) := N\hat{\sigma}^2 \log|\alpha_0 - c_{ij}B_{ij}| + \alpha_1 B_{ij} - \frac{1}{2}\alpha_2 B_{ij}^2 - \lambda w_{ij}|B_{ij}| \tag{11}$$

where c_{ij} denotes the (i,j)th co-factor of matrix $\mathbf{I}-\hat{\mathbf{B}}$, and $\{\alpha_l\}_{l=0}^2$ are defined as

$$\begin{aligned}
\alpha_0 &:= \det(\mathbf{I}-\hat{\mathbf{B}}) + c_{ij}\hat{B}_{ij}, \\
\alpha_1 &:= \left[\left(\mathbf{I}-\hat{\mathbf{B}}+\mathbf{e}_i\mathbf{e}_j^T\hat{B}_{ij}\right)\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T - \hat{\mathbf{F}}^{\text{new}}\tilde{\mathbf{X}}\tilde{\mathbf{Y}}^T\right]_{ij} \\
\alpha_2 &:= \|\tilde{\mathbf{Y}}^T\mathbf{e}_j\|_2^2
\end{aligned}$$

with $[\cdot]_{ij}$ representing the $(i,j)^{\text{th}}$ entry of the matrix between brackets. For numerical stability and computational savings, all co-factors c_{ij}, $j=1,\ldots N_g$, per row can be computed simultaneously by solving $(\mathbf{I}-\hat{\mathbf{B}})\mathbf{c}_i=\mathbf{e}_i$, with $\mathbf{c}_i := [c_{i1},\ldots,c_{iN_g}]^T$. After an iteration step is completed and $\hat{B}_{ij}^{\text{new}}$ is computed, $\mathbf{c}_i$ can be updated using the matrix inversion lemma as $\mathbf{c}_i = \mathbf{c}_i/(1+\hat{B}_{ij}^{\text{new}}-\hat{B}_{ij})$ before updating $\hat{B}_{ij}=\hat{B}_{ij}^{\text{new}}$.

A new estimate of B_{ij} is formed by maximizing $g_{ij}(B_{ij})$ in (11). To this end, consider two cases with $c_{ij}=0$ and $c_{ij}\neq 0$. If $c_{ij}=0$, the logarithmic term can be dropped from (11) yielding a standard Lasso problem with solution

$$\hat{B}_{ij}^{\text{new}} = \frac{\text{sign}(\alpha_1)}{\alpha_2}\max\{|\alpha_1|-\lambda w_{ij},0\}. \tag{12}$$

When $c_{ij}\neq 0$, three hypotheses are tested, namely: i) $B_{ij}>0$; ii) $B_{ij}=0$; and, iii) $B_{ij}<0$. For hypotheses i) and iii), the solution can be found in closed form after equating to zero the derivative of (11) w.r.t. B_{ij}. The roots found in both cases have to be tested against the corresponding hypothesis. Then, the surviving roots are grouped with $B_{ij}=0$ as candidate solutions, and the candidate yielding the maximum $g_{ij}(B_{ij})$ is the new estimate $\hat{B}_{ij}$.

Specifically, under hypothesis i) where $B_{ij}>0$, the derivative of $g_{ij}(B_{ij})$ in (11) takes the form $-N\sigma^2 c_{ij}/(\alpha_0-c_{ij}B_{ij})+(\alpha_1-\lambda w_{ij})-\alpha_2 B_{ij}$, which upon multiplication with $(\alpha_0-c_{ij}B_{ij})/c_{ij}$ turns into

$$\begin{aligned}
&-N\sigma^2+\alpha_1\frac{\alpha_0}{c_{ij}}-\lambda w_{ij}\frac{\alpha_0}{c_{ij}}-\left(\alpha_2\frac{\alpha_0}{c_{ij}}+\alpha_1-\lambda w_{ij}\right)B_{ij}+\alpha_2 B_{ij}^2 \\
&= p_0-\lambda w_{ij}\frac{\alpha_0}{c_{ij}}-(p_1-\lambda w_{ij})B_{ij}+\alpha_2 B_{ij}^2
\end{aligned} \tag{13}$$

under the definitions

$$\begin{aligned}
p_0 &:= -N\sigma^2+\alpha_1\frac{\alpha_0}{c_{ij}} \\
p_1 &:= -\alpha_1+\alpha_2\frac{\alpha_0}{c_{ij}}.
\end{aligned}$$

Consider the equation obtained by setting (13) equal to zero. If it has root(s), then they are given by

$$r_{ij}^+ = \frac{1}{2\alpha_2}\left[p_1-\lambda w_{ij}\pm\sqrt{(p_1-\lambda w_{ij})^2-4\alpha_2\left(p_0-\lambda w_{ij}\frac{\alpha_0}{c_{ij}}\right)}\right]. \tag{14}$$

Let B_{ij}^+ stand for the set containing the positive root(s) in (14). If the equation does not have a solution, B_{ij}^+ equals the empty set.

Similarly for hypothesis iii) where $B_{ij}<0$, setting the derivative of (11) equal to zero, one obtains an equation. If this equation has root(s), they are given by

$$r_{ij}^- = \frac{1}{2\alpha_2}\left[p_1+\lambda w_{ij}\pm\sqrt{(p_1+\lambda w_{ij})^2-4\alpha_2\left(p_0+\lambda w_{ij}\frac{\alpha_0}{c_{ij}}\right)}\right]. \tag{15}$$

Algorithm 1 : SML

```
Select the optimal value of ρ in (4), ρ_opt, via cross validation
Solve (4) with ρ_opt for F̃ and B̃
Estimate σ̂² as the sample variance of E=Ỹ − B̃Ỹ − F̃X̃
Compute weights w_ij = 1/[B̃]_ij, i, j = 1, ..., N_g
Compute Q(λ_max) via (S2) ∀i, j = 1, ..., N_g
Compute λ_max via (S9)
Select the optimal value of λ, λ_opt, via cross validation
for λ_l = λ_max, ..., λ_opt do
    Compute S_B(λ_l) via (S4)
    Initialize B̂ = B̃, F̂ = F̃, ε = 10^-4 and err = 10
    while err> ε do
        for i = 1,..., N_g do
            Obtain F̂^new by computing its row via (7) with b_i = b̂_i
        end for
        for i = 1,..., N_g do
            for j = 1,..., N_g do
                if B̂_ij ∉ S_B(λ_l) then
                    Compute cofactor of I − B̂, c_ij
                    if c_ij = 0 then
                        Compute B̂_ij^new via (12)
                    else
                        Compute B̂_ij^new via (16)
                    end if
                end if
            end for
        end for
        Compute err = ‖B̂ − B̂^new‖_F^2/‖B‖_F^2 + ‖F̂ − F̂^new‖_F^2/‖F‖_F^2
        Set B̂ = B̂^new and F̂ = F̂^new
    end while
    Compute Q_ij(λ_l) via (S1) ∀i, j = 1, ..., N_g
end for
Output B̂ and F̂.
```

Let B_{ij}^{-} denote the set containing the negative root(s) in (15). If the equation does not have a solution, B_{ij}^{-} becomes the empty set. Considering all three hypotheses, one arrives at

$$\hat{B}_{ij}^{\text{new}} = \underset{B_{ij} \in B_{ij}^{+} \cup B_{ij}^{-} \cup \{0\}}{\arg\max} \; g_{ij}(B_{ij}). \tag{16}$$

After a cycle is completed, the algorithm is checked for convergence by verifying whether the inequality $\|\hat{\mathbf{B}} - \hat{\mathbf{B}}^{\text{new}}\|_F^2 / \|\mathbf{B}\|_F^2 + \|\hat{\mathbf{F}} - \hat{\mathbf{F}}^{\text{new}}\|_F^2 / \|\mathbf{F}\|_F^2 < \varepsilon$ is satisfied, where ε is a prespecified small constant. If yes, the algorithm is stopped and $\hat{\mathbf{B}} = \hat{\mathbf{B}}^{\text{new}}$ and $\hat{\mathbf{F}} = \hat{\mathbf{F}}^{\text{new}}$ are output as the final estimates of $\mathbf{B}$ and $\mathbf{F}$; otherwise, $\hat{\mathbf{B}} = \hat{\mathbf{B}}^{\text{new}}$ and $\hat{\mathbf{F}} = \hat{\mathbf{F}}^{\text{new}}$ and one proceeds to execute the next cycle.

In order to increase the speed of the SML algorithm, the discarding rules proposed for sparse linear regression [66, 67] were adapted to the sparse SEM setup. Given λ, the discarding rules provide a means of computing a matrix $Q(\lambda)$, whose entries determining entries of $\mathbf{B}$ that can be set to zero *a priori* without be updated during the coordinate-ascent iterations. A detailed description of the discarding rules, together with the CV procedure to select the optimal λ, and the expression for the required $\lambda_{\max}$, that is, the minimum value of λ for which the solution to (3) is null, are provided in the supporting text S1.

SML algorithm

The overall SML approach described in the Methods section, including the ridge regression weights, the discarding rules, and the coordinate descent cycle is depicted step-by-step in Algorithm 1. The for-loop starting from line 8 and ending at the last line is the ℓ_1-regularized ML method for computing $\hat{\mathbf{B}}$ and $\hat{\mathbf{F}}$ in (3), which comprises the block coordinate ascent algorithm and discarding rules. In our computer program, these lines were written as a subroutine. Since the CV on line 7 needs to solve (3), the subroutine is also called on line 3 with λ varying from $\lambda_{\max}$ to $\lambda_{\min} = 10^{-4}\lambda_{\max}$. An additional subroutine implementing ridge regression was written to solve (4), and subsequently called on lines 1 and 2.

In the supporting text S1, three relevant extensions to the SML algorithm are described. First, stability selection [44] is applied to the SML, as an alternative to CV, to select the sparsity level so that the FDR is controlled. Second, the SML is extended to handle heteroscedasticity in the SEM error. Third, the SML is modified to enable inference of unknown eQTLs. In addition, supporting text S1 gives a description of the state-of-the-art AL-based and QDG algorithms that were considered for comparison with SML.

Acknowledgments

A preliminary version of the SML algorithm fully developed in this paper was presented at 2011 IEEE International Workshop on Genomic Signal Processing and Statistics, December 4-6, 2011, San Antonio, Texas, USA. We would like to thank Dr. Zhibin Chen in the Department of Microbiology and Immunology at the University of Miami for selecting genes used in the inference of the human gene network and for his help with interpreting the inferred network. We would also thank Anhui Huang at the University of Miami for his help with imputing the missing genotypes for the data used in the inference of the human gene network.

References

1. Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. Science 298: 799-804.

2. Butte AJ, Tamayo P, Slonim D, Golub TR, Kohane IS (2000) Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. Proc Natl Acad Sci USA 97: 12182-6.

3. Basso K, Margolin AA, Stolovitzky G, Klein U, Dalla-Favera R, et al. (2005) Reverse engineering of regulatory networks in human B cells. Nat Genet 37: 382-90.

4. Dobra A, Hans C, Jones B, Nevins JR, Yao G, et al. (2004) Sparse graphical models for exploring gene expression data. J Multivar Anal 90: 196-212.

5. Schäfer J, Strimmer K (2005) An empirical Bayes approach to inferring large-scale gene association networks. Bioinform 21: 754-764.

6. Friedman N, Linial M, Nachman I, Pe'er D (2000) Using Bayesian network to analyze expression data. J Comput Biol 7: 601-620.

7. Segal E, Shapira M, Regev A, Pe'er D, Botstein D, et al. (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nat Genet 34: 166-178.

8. Gardner TS, di Bernardo D, Lorenz D, Collins JJ (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. Science 301: 102-105.

9. di Bernardo D, Thompson MJ, Gardner TS, Chobot SE, Eastwood EL, et al. (2005) Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. Nat Biotechnol 23: 377-383.

10. Schäfer J, Strimmer K (2005) A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Stat Appl Genet Mol Biol 4: article 32.

11. Bonneau R, Reiss D, Shannon P, Facciotti M, Hood L, et al. (2006) The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets *de novo*. Genome Biol 7: R36.

12. Sima C, Hua J, Jung S (2009) Inference of gene regulatory networks using time-series data: a survey. Curr Genomics 10: 416-429.

13. Penfold CA, Wild DL (2011) How to infer gene networks from expression profiles, revisited. Interface Focus 1: 857-870.

14. Yip KY, Alexander RP, Yan KK, Gerstein M (2010) Improved reconstruction of *in silico* gene regulatory networks by integrating knockout and perturbation data. PLoS ONE 5: e8121.

15. Rockman MV (2009) Reverse engineering the genotype-phenotype map with natural genetic variation. Nature 456: 738-744.

16. Zhu J, Lum PY, Lamb J, GuhaThakurta D, Edwardsa S, et al. (2004) An integrative genomics approach to the reconstruction of gene networks in segregating populations. Cytogenet Genome Res 105: 363-374.

17. Zhu J, Wiener MC, Zhang C, Fridman A, Minch E, et al. (2007) Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations. PLoS Comput Biol 3: e69.

18. Zhu J, Zhang B, Smith EN, Drees B, Brem RB, et al. (2008) Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. Nat Genet 40: 854-61.

19. Kulp DC, Jagalur M (2006) Causal inference of regulator-target pairs by gene mapping of expression phenotypes. BMC Genet 7: 125.

20. Chen LS, Emmert-Streib F, Storey JD (2007) Harnessing naturally randomized transcription to infer regulatory relationships among genes. Genome Biol 8: R219.

21. Neto EC, Ferrara CT, Attie AD, Yandell BS (2008) Inferring causal phenotype networks from segregating populations. Genetics 179: 1089-1100.

22. Aten JE, Fuller TF, Lusis AJ, Horvath S (2008) Using genetic markers to orient the edges in quantitative trait networks: The NEO software. BMC Syst Biol 2: 34.

23. Millstein J, Zhang B, Zhu J, Schadt EE (2009) Disentangling molecular relationships with a causal inference test. BMC Genet 10.

24. Xiong M, Li J, Fang X (2004) Identification of genetic networks. Genetics 166: 1037-1052.

25. Liu B, de la Fuente A, Hoeschele I (2008) Gene network inference via structural equation modeling in genetical genomics experiments. Genetics 178: 1763-1776.

26. Logsdon BA, Mezey J (2010) Gene expression network reconstruction by convex feature selection when incorporating genetic perturbations. PLoS Comput Biol 6: e1001014.

27. Mi XJ, Eskridge K, Wang D (2010) Regression-based multi-trait QTL mapping using a structural equation model. Stat Appl Genet Mol Biol 9.

28. Gianola D, Sorensen D (2004) Quantitative genetic models for describing simultaneous and recursive relationships between phenotypes. Genetics 167: 1407-1424.

29. de los Campos G, Gianola D, Heringstad B (2006) A structural equation model for describing relationships between somatic cell score and milk yield in first-lactation dairy cows. J Dairy Sci 89: 4445-4455.

30. Wu XL, Heringstad B, Chang YM (2007) Inferring relationships between somatic cell score and milk yield using simultaneous and recursive models. J Dairy Sci 90: 3508-3521.

31. Jamrozik J, Bohmanova J, Schaeffer LR (2010) Relationships between milk yield and somatic cell score in canadian holsteins from simultaneous and recursive random regression models. J Dairy Sci 93: 1216-1233.

32. Valente BD, Rosa GJM, de los Campos G (2010) Searching for recursive causal structures in multivariate quantitative genetics mixed models. Genetics 185: 633-644.

33. Wu XL, Heringstad B, Gianola D (2010) Bayesian structural equation models for inferring relationships between phenotypes: a review of methodology, identifiability, and applications. J Anim Breed Genet 127: 3-15.

34. Rosa GJM, Valente BD, de los Campos G (2011) Inferring causal phenotype networks using structural equation models. Genet Sel Evol 43.

35. Neto EC, Keller MP, Attie AD, Yandell BS (2010) Causal graphical models in systems genetics: A unified framework for joint inference of causal network and genetic architecture for correlated phenotypes. Ann Appl Stat 4: 320-339.

36. Zou H (2006) The adaptive Lasso and its oracle properties. J Amer Stat Assoc 101: 1418-1429.

37. Tegner J, Yeung MK, Hasty J, Collins JJ (2003) Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling. Proc Natl Acad Sci USA 100: 5944-9.

38. Jeong H, Mason SP, Barabássi AL, Oltvai ZN (2001) Lethality and centrality in protein networks. Nature 411: 41-42.

39. Thieffry D, Huerta AM, Pérez-Rueda E, Collado-Vides J (1998) From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *Escherichia coli*. Bioessays 20: 433-440.

40. Tibshirani R (1996) Regression shrinkage and selection via the Lasso. J R Statistical Soc Ser B 58: 267–288.

41. Spirtes P, Glymour C, Scheines R (2000) Causation, Prediction, and Search. Cambridge, MA: MIT Press, 2 edition.

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

43. Li R, Tsaih SW, Shockley K (2006) Structural model analysis of multiple quantitative traits. PLoS Genet 2: e114.

44. Meinshausen N, Bhlmann P (2010) Stability selection. J R Statist Soc B 72: 417–473.

45. Pickrell JK, Marioni JC, Pai AA, Degner JF, Engelhardt BE, et al. (2010) Understanding mechanisms underlying human gene expression variation with RNA sequencing. Nautre 464: 768-772.

46. Frazer KA, Ballinger DG, Cox DR, Hinds DA, Stuve LL (2007) A second generation human haplotype map of over 3.1 million SNPs. Nature 449: 851-861.

47. Giannakis G, Mateos G, Farahmand S, Kekatos V, Zhu H (2011) Uspacor: Universal sparsity-controlling outlier rejection. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 1952–1955.

48. Howie BN, Donnelly P, Marchini J (2009) A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. PLoS Genet 5: e1000529.

49. Huang DW, Sherman BT, Lempicki RA (2009) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nat Protoc 4: 44-57.

50. Santiago T, Kulemzin SV, Reshetnikova ES, Chikaev NA, Volkova OY, et al. (2011) Fcrla is a resident endoplasmic reticulum protein that associates with intracellular igs, igm, igg and iga. Int Immunol 23: 43-53.

51. Wilson TJ, Gilfillan S, Colonna M (2010) Fc receptor-like a associates with intracellular igg and igm but is dispensable for antigen-specific immune responses. J Immunol 185: 2960-2967.

52. Chu CC, Paul WE (1997) An interleukin 4-induced mouse B cell gene isolated by cDNA representational difference analysis. Proc Natl Acad Sci USA 94: 2507-2512.

53. Chavana SS, Tiana W, Hsueha K, Jawaheerd D, Gregersend PK, et al. (2002) Characterization of the humanhomolog of the IL-4 induced gene-1. Proc Natl Acad Sci USA 1576: 7080.

54. Boulland ML, Marquet J, Molinier-Frenkel V, Mller P, Guiter C, et al. (2007) Human IL4I1 is a secreted l-phenylalanine oxidase expressed by mature dendritic cells that inhibits T-lymphocyte proliferation. Blood 110: 220-227.

55. Bollen KA (1989) Structural Equations with Latent Variables. Wiley-Interscience.

56. Kaplan D (2009) Structural Equation Modeling: Foundations and Extensions. Sage Publications, 2 edition.

57. Lee SY (2007) Structural Equation Modeling: A Bayesian Approach. Wiley.

58. Robert CP, Casella G (2004) Monte Carlo statistical method. Springer, 2 edition.

59. Carlin BP, Louis TA (2008) Bayesian Methods for Data Analysis. Chapman and Hall/CRC, 3 edition.

60. Holland JH (1972) Adaptation in Natural and Artificial Systems. Ann Arbor, MI: University of Michigan Press.

61. Goldberg DE (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley.

62. Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. J Stat Softw 33: 1-22.

63. Shipley B (2002) Cause and Correlation in Biology: A User's Guide to Path Analysis, Structural Equations and Causal Inference. Cambridge University Press.

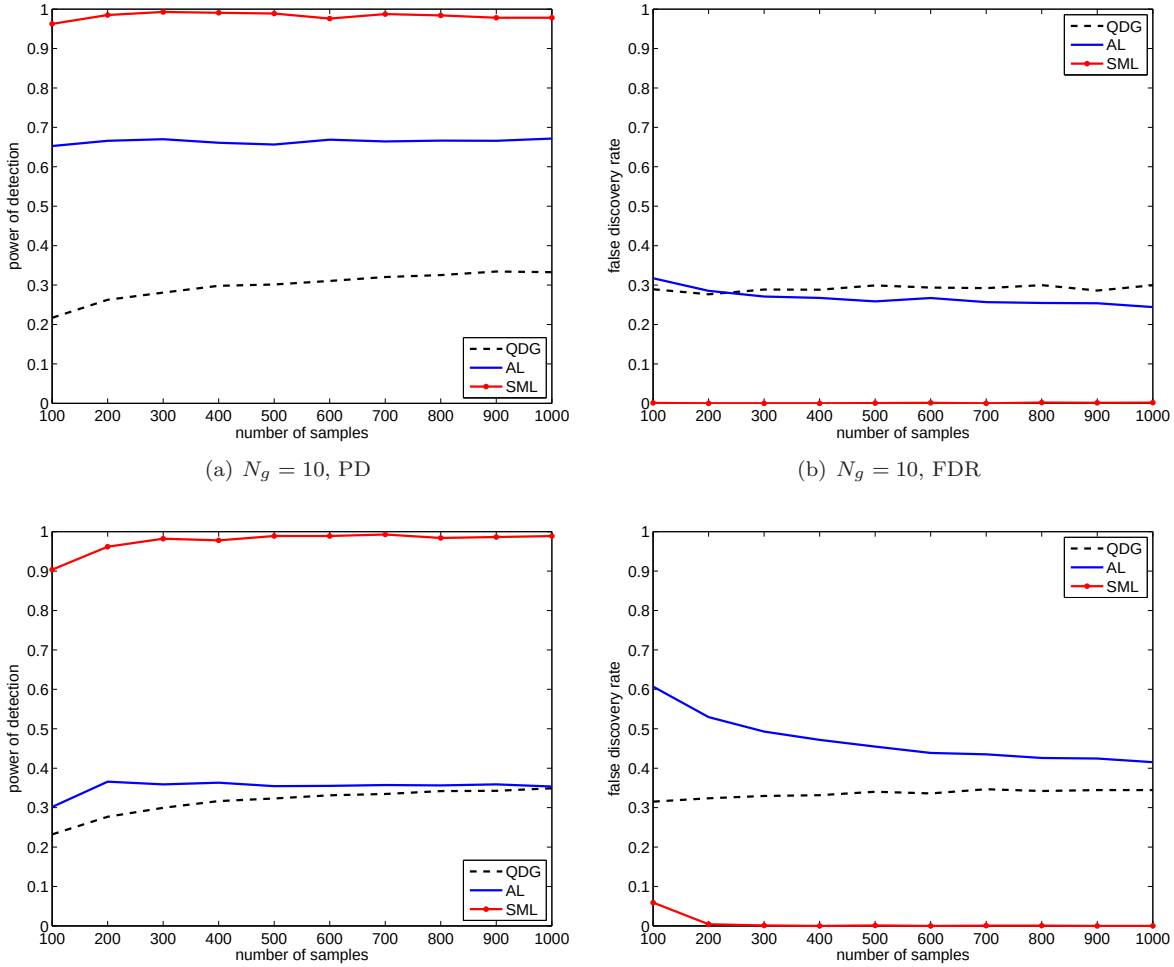
64. Pearl J (2009) Causality: Models, Reasoning, and Inference. Cambridge University Press, 2 edition.

65. Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer, 2 edition.

66. El Ghaoui L, Viallon V, Rabbani T (2010) Safe feature elimination in sparse supervised learning. Technical Report UC/EECS-2010-126, EECS Dept., University of California at Berkeley.

67. Tibshirani R, Bien J, Friedman J, Hastie T, Simon N, et al. (2012) Strong rules for discarding predictors in lasso-type problems. J R Statist Soc B 74: 245266.

Figure 1. Block diagram of the sparsity-aware maximum likelihood (SML) algorithm. The first and third blocks perform cross-validation to select optimal parameters ρ and λ to be used in (3) and (4), respectively. The third block produces weights $\{w_{ij}\}$ and error-variance estimate $\hat{\sigma}_e^2$ after solving (4). Finally, the fourth block takes data $\mathbf{X}$ and $\mathbf{Y}$ together with λ, $\{w_{ij}\}$ and $\hat{\sigma}_e^2$ and solves (3) to yield $\hat{\mathbf{B}}$, which represents the SML estimator for $\mathbf{B}$ in (1) revealing the genetic-interaction network. A more detailed description of the SML algorithm is given in Algorithm 1 in the Methods section.



(a) $N_g = 10$, PD

(b) $N_g = 10$, FDR

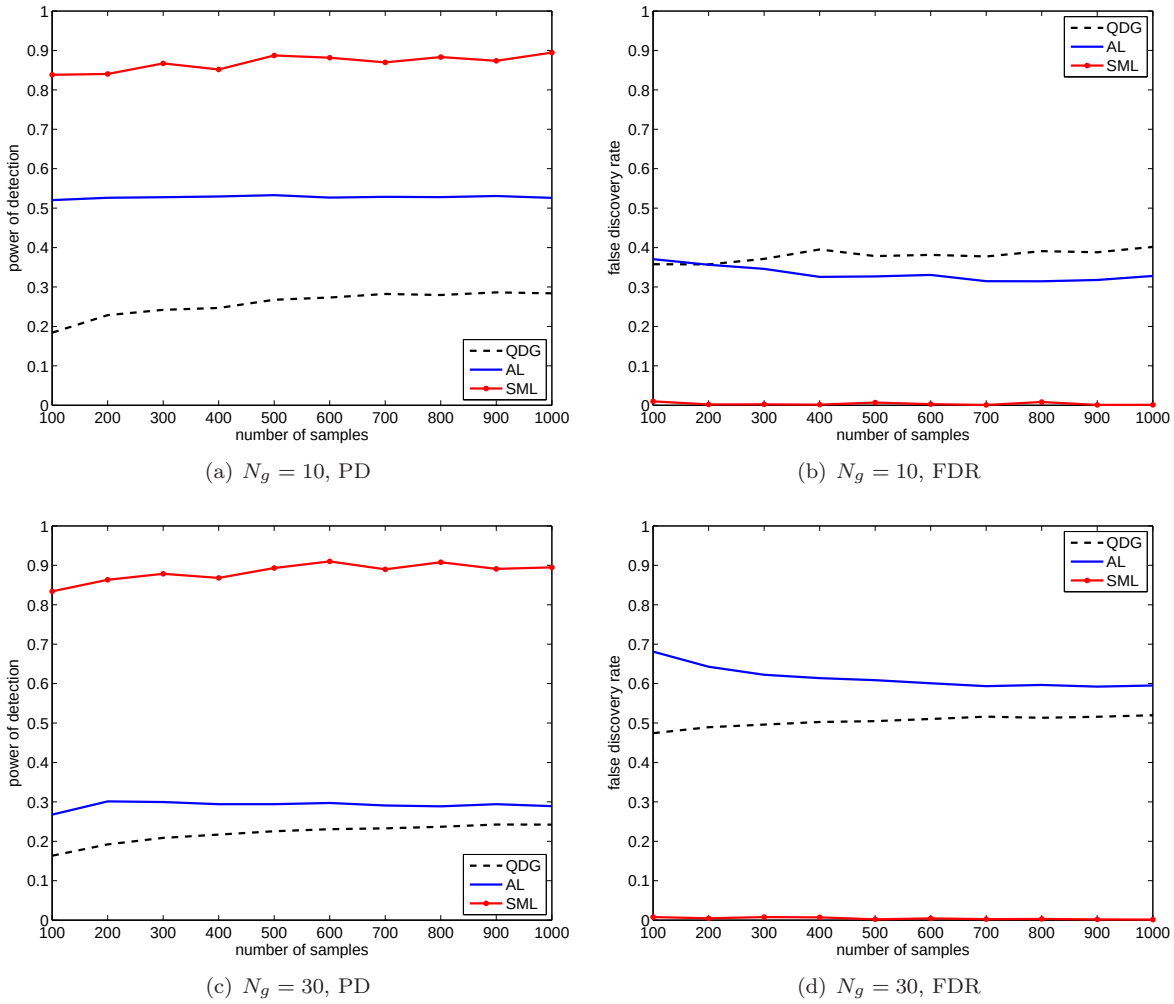
(c) $N_g = 30$, PD

(d) $N_g = 30$, FDR

Figure 2. Performance of SML, AL and QDG algorithms for directed *acyclic* **networks of $N_g = 10$ [(a) and (b)] or 30 [(c) and (d)] genes.** Expected number of nodes per node is $N_e = 3$. PD and FDR were obtained from 100 replicates of the network with different sample sizes (N= 100 to 1,000).

Table 1. Performance of SML, AL and QDG algorithms. Expected number of nodes per node is $N_e = 3$. PD and FDR were obtained from 100 replicates of the network with a sample size of 500.

Network	N_g	PD			FDR		
		SML	AL	QDG	SML	AL	QDG
DAG	10	0.9887	0.6564	0.3014	0.0007	0.2586	0.2991
	30	0.9891	0.3544	0.3232	0.0010	0.4548	0.3403
DCG	10	0.8872	0.5330	0.2677	0.0067	0.3268	0.3783
	30	0.8931	0.2941	0.2254	0.0020	0.6086	0.5047



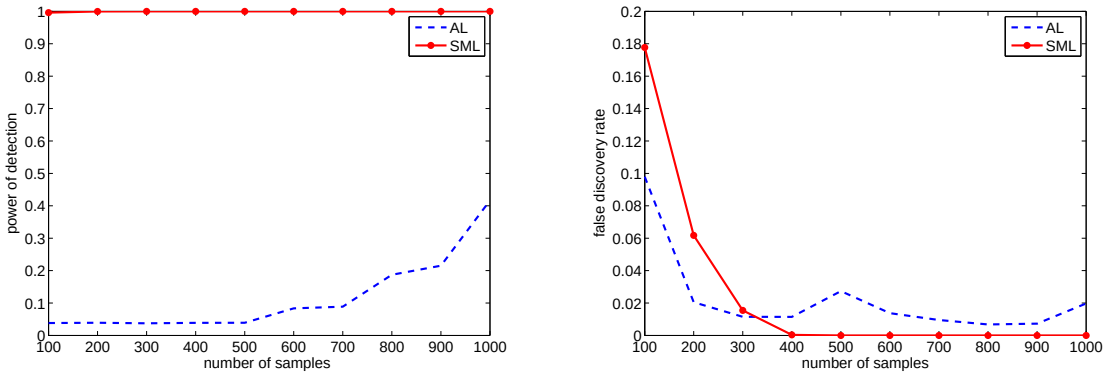
(a) $N_g = 10$, PD

(b) $N_g = 10$, FDR

(c) $N_g = 30$, PD

(d) $N_g = 30$, FDR

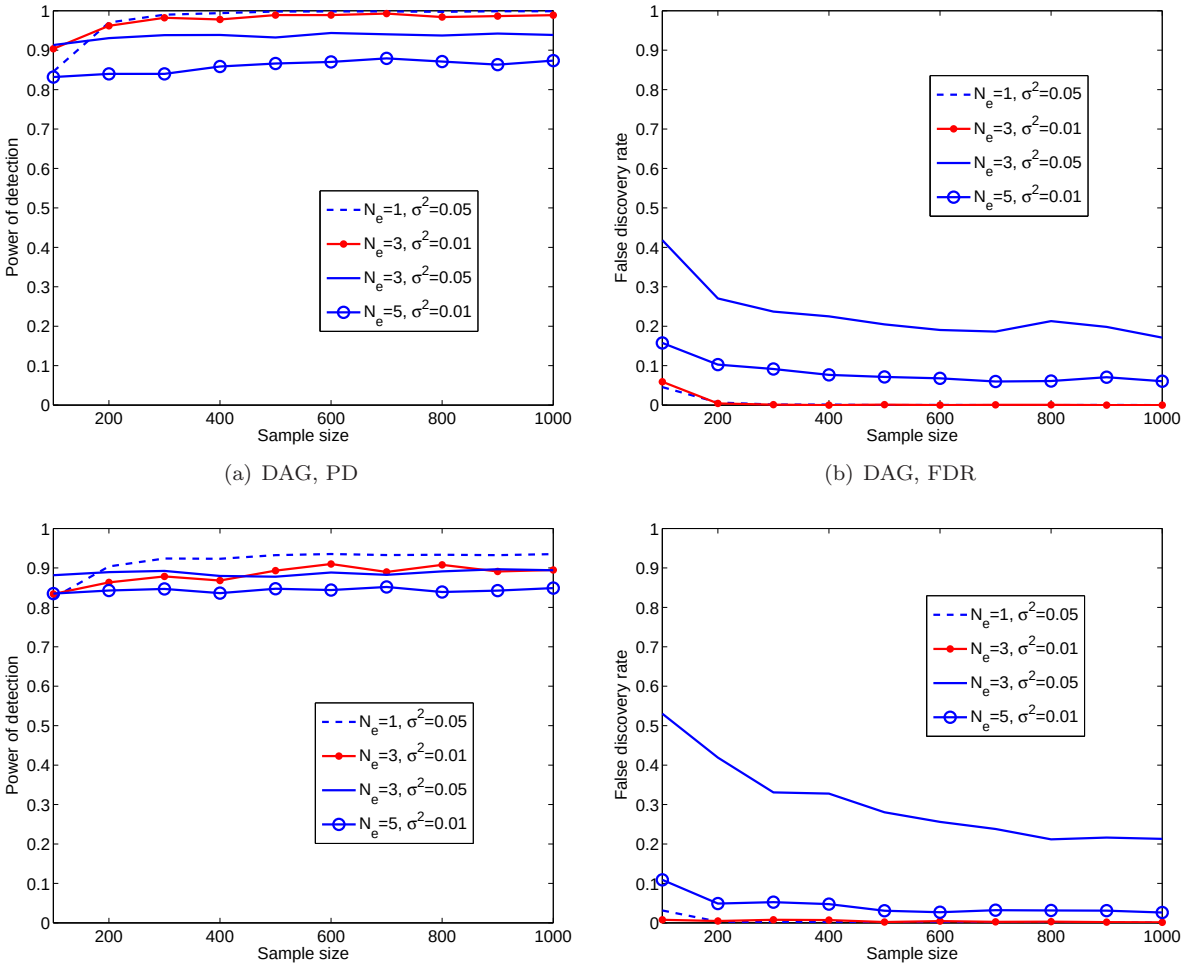
Figure 3. Performance of SML, AL and QDG algorithms for directed *cyclic* networks of $N_g = 10$ [(a) and (b)] or 30 [(c) and (d)] genes. Expected number of nodes per node is $N_e = 3$. PD and FDR were obtained from 100 replicates of the network with different sample sizes (N= 100 to 1,000).



(a) $N_g = 300$, PD

(b) $N_g = 300$, FDR

Figure 4. Performance of the SML and AL algorithms for directed *acyclic* **networks of** $N_g = 300$ **genes.** Expected number of nodes per node is $N_e = 1$. PD and FDR were obtained from 10 replicates of the network with different sample sizes (N= 100 to 1,000).



(a) DAG, PD (b) DAG, FDR

(c) DCG, PD (d) DCG, FDR

Figure 5. Performance of the SML algorithms for DAGs [(a) and (b)] or DCGs [(c) and (d)] of N_g=30 genes with an expected number of nodes per node $N_e \in$ {1,3,5} and error variance $\sigma^2 \in$ {0.01,0.05} . PD and FDR were obtained from 100 replicates of the network with different sample sizes (N= 100 to 1,000).

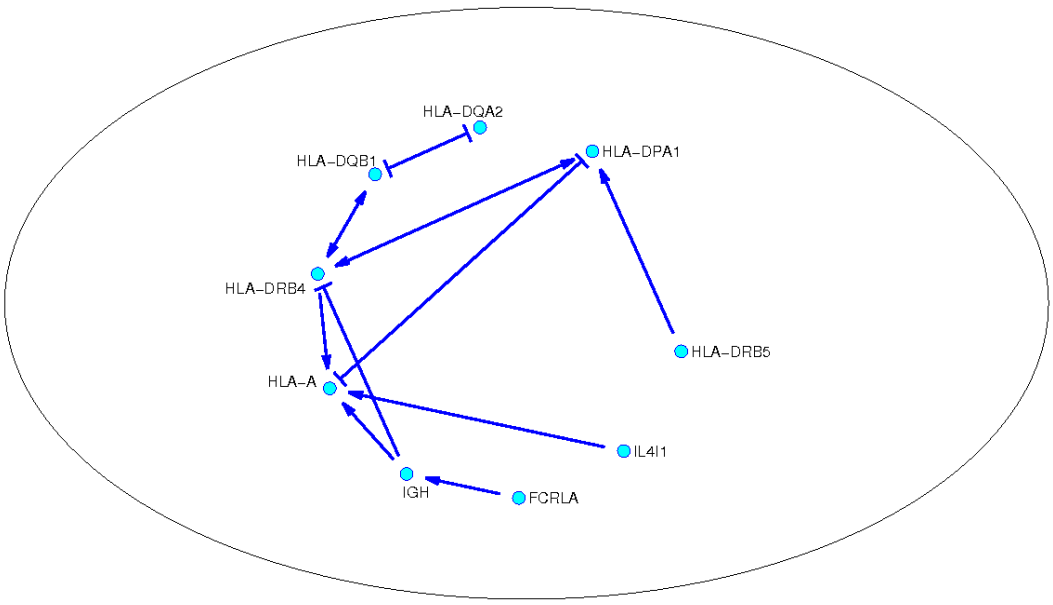


Figure 6. The network of 39 human genes inferred from gene expression and eQTL data with the SML algorithm. The 39 genes related to the immune function were chosen from [45] to have a reliable eQTL per gene. The SML algorithm was run with stability selection and edges were detected at an FDR < 0.1. See Table **??** for the IDs and description of 39 genes. IGH in this figure corresponds to gene ID ENSG00000211897. A $\dashv$ edge stands for inhibitory effect and a $\rightarrow$ edge stands for activating effect.