Structural factor equation models for causal network construction via directed acyclic mixed graphs

Yan Zhou*

Gilead Sciences, Foster City, California

**email:* yan.zhou4@gilead.com

and

Peter X.-K. Song*

Department of Biostatistics, University of Michigan, Ann Arbor, MI

**email:* pxsong@umich.edu

and

Xiaoquan Wen*

Department of Biostatistics, University of Michigan, Ann Arbor, MI

**email:* xwen@umich.edu

SUMMARY: Directed acyclic mixed graphs (DAMG) provide a useful representation of network topology with both directed and undirected edges subject to the restriction of no directed cycles in the graph. This graphical framework may arise in many biomedical studies, for example when a directed acyclic graph (DAG) of interest is contaminated with undirected edges induced by some unobserved confounding factors (e.g., unmeasured environmental factors). Directed edges in a DAG are widely used to evaluate causal relationships among variables in a network, but detecting them is challenging when the underlying causality is obscured by some shared latent factors. The objective of this paper is to develop an effective structural equation model (SEM) method to extract reliable causal relationships from a DAMG. The proposed approach, termed *structural factor equation model (SFEM)*, uses the SEM to capture the network topology of the DAG while accounting for the undirected edges in the graph with a factor analysis (FA) model. The latent factors in the SFEM enable the identification and removal of undirected edges, leading to a simpler and more interpretable causal network. The proposed method is evaluated and compared to existing methods through extensive simulation studies, and illustrated through the construction of gene regulatory networks related to breast cancer.

This is the author manuscript accepted for publication and has undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1111/biom.13322

Biometrics 000, 000–000 DOI: 000

000 0000

Key words: Directed acyclic graph; Factor analysis model; Network data; Regularization; Semi-Markov model.

1. Introduction

Reconstructing gene regulatory networks (GRN) using gene expression data furthers our understanding of gene function and cellular dynamics in biological systems by elucidating regulatory mechanisms. Graphical models are a popular tool to analyze and visualize conditional independence among variables of interest. A graphical model includes nodes representing random variables and edges encoding relationships between the enclosing nodes. Graphical models are classified into two classes depending on whether edges are directional: directed graphical models and undirected graphical models. A directed graphical model, also known as a Bayesian network, is a graphical model whose dependence structure is represented by a directed acyclic graph (DAG). For example, a directed edge between two genes may represent a molecular chain reaction of one gene regulating the other. The utility of DAGs for inferring causality has received much attention in the reconstruction of gene regulatory networks (Friedman et al., 2000; Segal et al., 2003; Hartemink et al., 2002; Pe'er et al., 2001).

When a DAG is used for causal network inference, some of the directed edges are often masked by undirected edges induced by unmeasured confounding variables. The resulting graph may become a mixed graph with both directed and undirected edges, or even an undirected graph (Anandkumar et al., 2013). Thus, it seems inevitable to invoke a more general graph than a DAG to analyze the underlying network topology, in which undirected edges are allowed. This motivates us to focus on an analysis of causal relationships in a directed acyclic mixed graph (DAMG). DAMG is sometimes called an acyclic directed mixed graph, and also known as a semi-Markov model. It contains both directed and undirected edges, subject to the restriction of no directed cycles in the graph. In a DAMG, the set of directed edges represents the causal relationships between nodes and constitutes the DAG of interest. It can, however, be contaminated by undirected edges introduced by some unobserved factors. In practice, the latent factors may include, for example, biomarkers that

are not included in experimental chips, environmental variables, and underlying populations among experimental samples. Unfortunately, such shared masking factors are often not directly measured in experiments despite their potential influence on measurements. In the literature, methods for removing these masking factors have not been systematically investigated. Here, we propose a new method that identifies and removes nuisance undirected edges in the reconstruction of causal relationships to obtain an interpretable causal network.

Learning the dependence structure of a DAG from data presents a significant challenge because the number of candidate DAGs can grow super-exponentially along with the number of nodes (Robinson, 1973). There are three types of approaches to learning DAG structures: search-and-score approaches, constraint-based approaches, and hybrid approaches. A search-and-score approach attempts to learn a DAG structure by optimizing some criteria, such as the BIC or validation set likelihood, using either a search algorithm (Lam and Bacchus, 1994; Heckerman et al., 1995) or Bayesian posterior distribution (Friedman and Koller, 2003; Ellis and Wong, 2008; Zhou, 2011). A constraint-based approach tries to prune a set of possible edges identified by conditional independence hypothesis tests, including the well-known Peter-Clark (PC) algorithm (Spirtes et al., 2000), or by removing conditional dependencies that fall below a threshold (Cheng et al., 2002). A constraint-based method is developed to prune a set of edges with a focus of improving computational efficiency (Li and Yang, 2004; Tsamardinos et al., 2006).

A vast majority of recent work has focused on the reconstruction of a sparse DAG through a penalized likelihood approach. In the special case where a topological ordering of the nodes is given, learning the structure of a DAG is equivalent to sparse estimation of the modified Cholesky decomposition of a concentration matrix (i.e., the inverse of the corresponding covariance matrix), which is computationally feasible; see for example, Li and Yang (2005); Huang et al. (2006); Levina et al. (2008); Shojaie and Michailidis (2010), among others.

The information on node ordering is usually determined by a natural ordering of temporal observations, previous experiments, and *a priori* knowledge (Shojaie and Michailidis, 2010). For example, when learning GRNs for microarray data, *a priori* knowledge of the node ordering could be obtained from the existing annotation software such as Cytoscape (Lopes et al., 2010). If there is no established known node ordering, some penalized score-based methods (e.g., Fu and Zhou (2013) and Aragam and Zhou (2015)) may be first applied to estimate DAG structures, which is done without *a priori* knowledge of node ordering, followed by extracting the node ordering from the estimated DAG.

To fill in the technical gap where no systematic work is available to assess sparse causal relationships in DAMGs, we develop a regularization estimation method to extract and evaluate a sparse causal network in the form of a DAG. We develop a new method based on the structural factor equation model (SFEM) introduced in detail in Section 2 with conditions for model identifiability. Section 3 concerns the penalized estimation of DAGs based on an EM-Coordinate-Descent (EM-CD) algorithm for numerical implementation. Operating characteristics of the proposed method are examined on both simulated and real data in Sections 4 and 5. We conclude with a discussion in Section 6 in which we discuss the estimation of directed acyclic graphs with unknown node ordering.

2. Structural factor equation model

2.1 *Model*

Given a P-dimensional random vector $\mathbf{y} = (y_1, \ldots, y_P)^T$ with known variable ordering, we use a DAG $\mathcal{G} = (V, E)$ to describe causal relations, where V is the set of vertices (or variables or nodes) and E is the collection of edges. That is, each variable y_i corresponds to one node in the DAG, and a directed edge between two nodes indicates a causal relationship between them. Without loss of generality, we assume that $\mathbf{y}$ has been sorted according to

its known ordering, which means a causal relationship is only possible from variable y_j to variable y_i, denoted by $y_j \to y_i$, for $j < i$. The set of parental nodes of y_i is denoted by $pa(i) = \{j : j < i, y_j \to y_i\}$. Specifically, for any $k < i$, if $k \notin pa(i)$ then y_i is independent of y_k conditioning on $\{y_j\}_{j \in pa(i)}$.

To model causality among the components of $\mathbf{y}$, we invoke a structural equation model (SEM): $y_i = \sum_{j \in pa(i)} \theta_{ij} y_j + \epsilon_i,\ i = 1, \ldots, P$, where ϵ_i's are normal random errors with mean 0 and independent of y_i's parental nodes. Regression parameter θ_{ij} is a coefficient representing the association of y_i with y_j, conditional on all other parental nodes of y_i. The matrix form of the SEM is: $\mathbf{y} = \Theta\mathbf{y} + \boldsymbol{\epsilon}$, where the vector of errors $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_P)^T$ has mean 0 and covariance W. Here $\Theta = \{\theta_{ij}\}$ is a $P \times P$ lower triangular matrix with zeros on the diagonal, and is termed the *weighted adjacency matrix* of the DAG $\mathcal{G}$. Given both Θ of DAG $\mathcal{G}$ and W, the first two moments of the SEM are $\boldsymbol{\mu} = E(\mathbf{y}) = 0$ and $\Sigma = \mathrm{Cov}(\mathbf{y}) = (I - \Theta)^{-T} W (I - \Theta)^{-1}$, which are uniquely determined by the two matrices Θ and W. This formulation requires *a priori* variable ordering.

We propose to model the covariance W by the classical factor analysis model (FAM): $W = BB^T + \Psi$, where B is a $P \times K$ factor loading matrix for K ($\leqslant P$) latent factors and Ψ is a $P \times P$ diagonal matrix of uniqueness (Johnson and Wichern, 2007). Combining the SEM and FAM models leads to the following structural factor equation model (SFEM):

$$\mathbf{y} = \Theta\mathbf{y} + B\mathbf{z} + \mathbf{e}, \tag{1}$$

where $\mathbf{z}$ is a K-variate vector of uncorrelated latent factors following multivariate normal distribution $\mathrm{MVN}_K(0, I)$ and $\mathbf{e}$ is an error vector distributed according to $\mathrm{MVN}_P(0, \Psi)$ and is independent of $\mathbf{z}$. Moreover, the first two moments of $\mathbf{y}$ are, respectively, $\boldsymbol{\mu} = 0$ and $\Sigma = (I - \Theta)^{-T}(BB^T + \Psi)(I - \Theta)^{-1}$. It is obvious that SFEM in (1) reduces to the classical SEM when $K = 0$. From (1), we can also see that conditioning on the vector of K unobserved latent variables $\mathbf{z}$, the vector of variables $\mathbf{y}$ satisfies the SEM for a DAG. Our objective is to

estimate the weighted adjacency matrix Θ of interest, the loading matrix B and uniqueness Ψ, as well as to determine the number of factors K.

2.2 *Graphical representation of SFEM*

Due to the potential influence of latent factors on causal relationships, we consider a more general graphical model than a DAG to accommodate undirected edges. We consider the class of directed mixed graphs (DMGs) that contain both directed and undirected edges. A mixed graph is defined by $\mathcal{G} = (V, E, U)$, where V is a finite set of vertices and $E, U \subseteq V \times V$ are two disjoint sets of edges. The edges in E are directed; that is, $(i, j) \in E \Rightarrow (j, i) \notin E$, denoted by $i \to j$. The edges in U are undirected or bi-directed; that is, $(i, j) \in E \Rightarrow (j, i) \in E$ and *vice versa*, denoted by $i \leftrightarrow j$. Part A of Figure 1 displays four examples of DMGs. The DMG shown in panel A(b) is cyclic, a type of DMG that is not considered in this paper.

[Figure 1 about here.]

In this paper, we focus on directed acyclic mixed graphs (DAMGs), a subclass of DMGs that do not include directed cycles. More specifically, a DAMG (V, E, U) consists of two subgraphs: one is a DAG (V, E) consisting of all directed edges, which is captured by a weighted adjacency matrix Θ; and the other is a subgraph containing all undirected edges (V, U), which are obtained by nonzero entries in the covariance matrix $W = BB^T + \Psi$ with $W_{ij} = W_{ji} \neq 0$ for $(i, j) \in U$ or $i = j$. In the gene regulatory network study, common factors attributed to matrix B could include, for example, environmental variables, which are not measured but may alter gene expressions substantially. These factors are useful to explain the mechanism of generation of undirected edges that contaminate the underlying causal relationships of interest. For example, Figure 1 A(c) and A(d) show that the directed chain networks among nodes Y_1, Y_2 and Y_3 (which contains a subgraph of interest, namely Figure 1 A(a), shown by the arrowed solid edges) is masked by undirected edges (indicated by dashed lines). Intuitively, it would be impossible to reconstruct a DAG (i.e., the chain graph

in panel A(a)) if these nuisance undirected edges were not properly removed. Our strategy is to identify and quantify potential triggers of undirected edges via the factor model, as illustrated in Part B of Figure 1. Figure 1B shows an example in which undirected edges arise from three shared common latent factors z_1, z_2 and z_3 among the 9 measured variables $y_1, \ldots, y_9$; marginalizing these latent factors will lead to many nuisance undirected edges in a complex DAMG. The proposed SFEM is developed to identify and reconstruct this DAG by conditioning out the three latent triggers responsible for the nuisance edges.

2.3 *Parameter identifiability in SFEM*

The parameters in the SFEM (1) include a lower triangular $P \times P$ weighted adjacency matrix Θ, a $P \times K$ factor loading matrix B and a diagonal $P \times P$ uniqueness matrix Ψ. The SFEM (1) may be rewritten as

$$\mathbf{y} = (I - \Theta)^{-1} B\mathbf{z} + (I - \Theta)^{-1}\mathbf{e} = \Gamma\mathbf{z} + \boldsymbol{\delta}, \tag{2}$$

where $\Gamma = (I - \Theta)^{-1}B$ and $\boldsymbol{\delta} = (I - \Theta)^{-1}\mathbf{e}$. The resulting covariance matrix of $\mathbf{y}$ is $\Sigma = \Gamma\Gamma^T + \Sigma_\delta$ with $\Sigma_\delta = (I-\Theta)^{-1}\Psi(I-\Theta)^{-T}$. It is well known that the factors and loadings are not separably identified without further restrictions. Note that the factors $\mathbf{z} \sim \text{MVN}_K(0, I)$ and loadings B enter the likelihood through $\Gamma\Gamma^T$. Hence, for any $K \times K$ rotation matrix Π, we have $\Gamma\Gamma^T = \Gamma\Pi\Pi^T\Gamma^T$, producing observationally equivalent models. Thus, we impose the following regularity conditions in order to identify parameters in both Σ_δ and Γ in model (2).

- Condition (A): Assume that $\Gamma^T\Sigma_\delta^{-1}\Gamma = B^T\Psi^{-1}B$ is diagonal with distinct entries arranged in a decreasing order.
- Condition (B): Assume that there exists a unique modified Cholesky decomposition of $\Sigma_\delta = (I - \Theta)^{-1}\Psi(I - \Theta)^{-T}$.

Condition (A) is a usual restriction for maximum likelihood estimation (MLE) in factor

analysis (see e.g., Lawley and Maxwell (1962), Bai and Li (2012)). This condition is needed to ensure that the reparametrization does not affect the decomposition of the total variance into a sum of loadings B and uniqueness Ψ. In other words, it ensures that solutions from the MLE obtained under the reparametrization can be uniquely transformed back to the original parametrization. Condition (B) is required to prohibit the arbitrary permutation of node ordering, so that the solution from the algorithm is unique. By taking $\Sigma_\delta^{-1} = (I - \Theta)^T \Psi^{-1} (I - \Theta)$, we obtain an alternative estimator to the classic SEM. The fact that the DAG representation (Θ, Ψ) encodes more conditional independence relations than the inverse covariance matrix Σ_δ^{-1} motivates us to obtain Θ for a simple and interpretable causal network.

3. Regularized estimation

3.1 *Formulation*

Regularization methods are appealing in network learning settings because the dimension of unknown parameters (e.g., entries in Θ) can quickly exceed the sample size of the data. When a natural ordering of the variables is available, and the number of latent factors $K = 0$ (i.e., $B = 0$), the reconstruction of a sparse DAG is equivalent to the sparse estimation of the modified Cholesky decomposition of Σ_δ^{-1}. In this case, the identifiability condition (A) automatically holds. Several regularization approaches have been proposed to shrink elements in Θ to zero. See Pourahmadi (1999); Wu and Pourahmadi (2003); Bickel and Levina (2008); Huang et al. (2006); Levina et al. (2008), just to name a few. More specifically, Huang et al. (2006) proposed adding an L_1 norm penalty on Θ to encourage zeros. Levina et al. (2008) proposed a banding procedure using a nested LASSO penalty. Recently, Shojaie and Michailidis (2010) employed the adaptive LASSO penalty to estimate the skeleton of a DAG in SEMs and showed that the LASSO method is not sensitive to random permutations of the order of variables in $\mathbf{y}$.

Given N samples $\mathbf{y}_n = (y_{n1}, \ldots, y_{nP})^T, n = 1, \ldots, N$, we want to detect the sparse skeleton of a DAG adjusting for latent factors. We propose the following penalized loss function:

$$\min_{\Theta} \frac{1}{2N} \sum_{n=1}^{N} (\mathbf{y}_n - \Theta \mathbf{y}_n)^T (BB^T + \Psi)^{-1} (\mathbf{y}_n - \Theta \mathbf{y}_n) + \lambda \sum_{i=1}^{P} \sum_{j=1}^{i-1} \xi_{ij} |c_{ij} \theta_{ij}|, \tag{3}$$

where λ is a nonnegative tuning parameter, c_{ij} represents the prior causal relationship of y_j on y_i, and ξ_{ij} is the adaptive weights of y_j on y_i. The L_1 norm penalty term in the above loss function (3) regularizes the sparsity in Θ.

Prior knowledge on the existence of causal relationships in $\mathbf{y}$ can be incorporated into the regularization procedure through a pre-specified $P \times P$ flag matrix $C = \{c_{ij}\}$, whose (i, j)-th element is given by:

$$c_{ij} = \begin{cases} 1 \text{ if there is no prior information of causality between } j \text{ and } i, \text{ when } j < i; \\ 0 \text{ if there exists prior knowledge of causality } j \to i, \text{when } j < i. \end{cases} \tag{4}$$

Matrix C in the penalty function is useful for screening all available edges in exploratory analyses. In addition, $\Xi = \{\xi_{ij}\}$ is a $P \times P$ lower triangular matrix of adaptive weights with the (i, j)-th element given by

$$\xi_{ij} = \begin{cases} \max(1, |\widetilde{\theta}_{ij}|^{-\gamma}), & \text{if } c_{ij} = 1 \text{ and } j < i; \\ 0, \text{ otherwise}, \end{cases} \tag{5}$$

where $\widetilde{\theta}_{ij}$ is the estimate of θ_{ij} obtained from classical LASSO estimation given by (3) with $\xi_{ij} = 1$ if $c_{ij} = 1$ and $j < i$.

3.2 *EM-Coordinate-Descent Algorithm*

We propose a two-step iterative approach to estimate three unknown matrices (Θ, B, Ψ). Given the current estimates $(B^{(t)}, \Psi^{(t)})$, $\Theta^{(t+1)}$ is updated by minimizing the penalized loss function (3) using the coordinated descent (CD) algorithm, and then $(B^{(t+1)}, \Psi^{(t+1)})$ are updated through the EM algorithm. Both the EM and CD algorithms are discussed below. Repeating the two-step procedure iteratively until convergence yields estimates $(\hat{\Theta}, \hat{B}, \hat{\Psi})$.

EM Algorithm. We use the EM algorithm to estimate (B, Ψ) in the factor analysis model. We mcan implement the EM algorithm by treating the latent factors $\mathbf{z}_n = (z_{n1}, \ldots, z_{nK})^T$, $n = 1, \ldots, N$ as "missing data" and Θ as a fixed "known" constant matrix. The M-step maximizes the log-likelihood of the full data $\{(\mathbf{y}_n^* \triangleq \mathbf{y}_n - \Theta\mathbf{y}_n,\ \mathbf{z}_n),\ n = 1, \ldots, N\}$. We outline the EM algorithm to update B and Ψ respectively at the $(t+1)$-th iteration. In the E-step, we obtain the following moments of $\mathbf{z}_n$, $n = 1, \ldots, N$,

$$\begin{aligned} E(\mathbf{z}_n|\mathbf{y}_n^*; B, \Psi) &= (B^T\Psi^{-1}B + I_K)^{-1}B^T\Psi^{-1}\mathbf{y}_n^*, \\ Var(\mathbf{z}_n|\mathbf{y}_n^*; B, \Psi) &= I_K - B^T\Psi^{-1}B(B^T\Psi^{-1}B + I_K)^{-1}, \\ E(\mathbf{z}_n\mathbf{z}_n^T|\mathbf{y}_n^*; B, \Psi) &= E(\mathbf{z}_n|\mathbf{y}_n^*; B, \Psi)E(\mathbf{z}_n^T|\mathbf{y}_n^*; B, \Psi) + Var(\mathbf{z}_n|\mathbf{y}_n^*; B, \Psi). \end{aligned} \tag{6}$$

In the M-step, B and Ψ are updated at the $(t+1)$-th iteration by, respectively,

$$\begin{aligned} B^{(t+1)} &= \Big\{\sum_{n=1}^N \mathbf{y}_n^* E(\mathbf{z}_n^T|\mathbf{y}_n^*; B^{(t)}, \Psi^{(t)})\Big\}\Big[\sum_{n=1}^N \Big\{E(\mathbf{z}_n\mathbf{z}_n^T|\mathbf{y}_n^*; B^{(t)}, \Psi^{(t)})\Big\}\Big]^{-1}, \\ \Psi^{(t+1)} &= \frac{1}{N}\sum_{n=1}^N E\Big\{(\mathbf{y}_n^* - B\mathbf{z}_n)(\mathbf{y}_n^* - B\mathbf{z}_n)^T|\mathbf{y}_n^*; B^{(t)}, \Psi^{(t)}\Big\}. \end{aligned} \tag{7}$$

Noting that Ψ takes the special form $\Psi = \sigma^2 I_P$, we consider a simple re-parameterization by letting $\tilde{B} = \sigma^{-1}B$ and $\tilde{\mathbf{z}}_n = \sigma\mathbf{z}_n$. Clearly, $B\mathbf{z}_n$ and $\tilde{B}\tilde{\mathbf{z}}_n$ follow the same distribution. Thus, the EM algorithm updates $\tilde{B}$ and σ^2 by the following expressions:

$$\begin{aligned} \tilde{B}^{(t+1)} &= \Big\{\sum_{n=1}^N \mathbf{y}_n^* E\Big(\tilde{\mathbf{z}}_n^T|\mathbf{y}_n^*; \tilde{B}^{(t)}, \sigma^{2(t)}\Big)\Big\}\Big\{\sum_{n=1}^N E\Big(\tilde{\mathbf{z}}_n\tilde{\mathbf{z}}_n^T|\mathbf{y}_n^*; \tilde{B}^{(t)}, \sigma^{2(t+1)}\Big)\Big\}^{-1}, \\ \sigma^{2(t+1)} &= \frac{1}{NQ}\sum_{n=1}^N \mathbf{y}^{*T}_n\Big\{I_Q - \tilde{B}^{(t)}\big(I_K + \tilde{B}^{T(t)}\tilde{B}^{(t)}\big)^{-1}\tilde{B}^{T(t)}\Big\}\mathbf{y}_n^*. \end{aligned} \tag{8}$$

Coordinate Descent Algorithm. We implement a coordinate descent algorithm to obtain the optimal solution that minimizes the L_1-norm penalized loss function (3) under a fixed positive definite matrix $W = BB^T + \Psi$. The sparse solution of Θ is obtained efficiently by an active-shooting algorithm. Refer to Section 1 of the Supporting Information for details.

EM-CD Algorithm. Finally, combining the EM and CD algorithms, termed the EM-CD algorithm, we can iteratively update Θ, B and Ψ as follows:

Step 1. Initialization of $B^{(0)}$, $\Psi^{(0)}$, and $\boldsymbol{\beta}^{(0)}$ with some suitable values.

Step 2. Given $(B^{(t)}, \Psi^{(t)}, \Theta^{(t)})$, update $\Theta^{(t+1)}$ by the CD active-shooting algorithm.

Step 3. Given $\Theta^{(t+1)}$, update $(B^{(t+1)}, \Psi^{(t+1)})$ via the EM algorithm until convergence.

Step 4. Repeat steps 2-3 above until convergence.

3.3 *Tuning parameter selection*

Choosing the number of latent factors K and tuning the sparsity parameter λ are both of critical importance in the proposed method. Since K can affect the resulting sparsity in the estimated Θ, it has to be tuned properly. In the factor analysis model literature, some methods have been developed for selecting K. For example, Bai and Ng (2002) and Onatski (2010) proposed methods to determine the number of factors in certain approximate factor analysis models. Onatski (2009) developed test statistics for a hypothesized number of factors using the empirical distribution of eigenvalues of the sample covariance matrix. Hirose and Konishi (2012) and Caner and Han (2014) employed shrinkage estimation to determine relevant factors. Here, we invoke an "eigenvalue ratio (ER)" criterion (Ahn and Horenstein, 2013) to select K, mainly for its simplicity and computational ease. In a general factor model given in (2), we can convert the selection of K in the original loading matrix B to that in the new loading matrix Γ. Following Ahn and Horenstein (2013), for a sample covariance matrix $YY^T/(NP)$, denote its k^{th} largest eigenvalues by η_k, $k = 1, \ldots, \min(N, P)$. The corresponding eigenvalue ratio is given by $\text{ER}(k) = \eta_k/\eta_{k+1}$. The eigenvalue ratio criterion is given by $K_{\text{ER}} = \arg\ \max_{K_{min} \leqslant k \leqslant K_{max}}\ \text{ER}(k)$, where K_{min} and K_{max} may be prespecified according to the scree plot, *say*, $K_{min} = 1$ or 2 and $K_{max} = min(N, P)/2$.

To select tuning parameter λ, we adopt M-fold cross-validation. Since the true model is believed to be sparse, we use the ordinary least squares (OLS) estimates instead of the shrunken estimates to calculate the cross-validation error score. This is because the cross-validation error score based on the shrunken estimates often leads to severe false positive rates when there are many potential poor predictors (Peng et al., 2010; Efron et al., 2004).

The OLS estimates are suggested in the literature as a reasonable remedy, as confirmed in our simulation studies. The Bayesian information criterion (BIC), another popular tuning parameter selection method, is not considered here mainly because estimating the degrees of freedom required by the BIC is difficult under a nonorthogonal design.

4. Simulation Experiments

4.1 *Simulation setup*

To examine the performance of the proposed SFEM for the estimation of a sparse DAG in DAMGs, we consider two types of DAG designs.

In simulation experiment I, we consider a small and simple DAG with $P = 50$ nodes and $M = 25$ edges that is randomly generated by the R-package `pcalg` (Kalisch and Bühlmann, 2007). To control for the sparsity of the DAG, we set the maximum number of parents for each node at 2, and the depth of the DAG to 3, and then randomly generate DAGs until the exact number of $M = 25$ edges is achieved.

In simulation experiment II, we consider a more complex DAG consisting of 19 master regulators (i.e., parental nodes). Among them, 4 are strong master regulators, each influencing 14 to 18 nodes, 7 are moderate master regulators, each influencing 3 to 7 nodes, and the rest are 8 weak parental nodes that link to only 1 or 2 offspring nodes. Such a DAG is generated by first randomly selecting 19 master parental nodes, and then further randomly selecting offspring nodes within each parental node. As a result, we create a DAG with $M = 100$ edges. In this second experiment we vary both the number of nodes and the number of latent factors. We set up the SFEM with fixed $P = 200$ nodes and a varying number of latent factors $K = 1, 5, 10$, and also set up the SFEM with fixed $K = 5$ but varying number of nodes $P = 50, 100, 200$. Clearly, with a fixed number of edges $M = 100$, a larger number of nodes P leads to a sparser network.

In both simulation designs we generate $N = 25$ and 100 units of networks, respectively, from the specified SFEMs above. In addition, we generate the elements of the weighted adjacency matrix Θ by $\theta_{ij} \overset{i.i.d.}{\sim} U([-3,-1]\bigcup[1,3])$ in Simulation I, and set constant $\theta_{ij} = 0.5$ in Simulation II. In each case, we simulate latent factors $z_{nk} \overset{i.i.d.}{\sim} N(0,1)$, loadings $B_{ik} \overset{i.i.d.}{\sim} U([-b,-a]\bigcup[a,b])$ and noise $e_{nj} \overset{i.i.d.}{\sim} N(0,\sigma^2)$, where the parameters a, b and σ^2 are chosen to satisfy a pre-specified percent of explained variability (PEV): $PEV = \sqrt{tr(\Sigma_\delta)/tr(\Sigma)}$, where $\Sigma_\delta = (I-\Theta)^{-1}\Psi(I-\Theta)^{-T}$ and $\Sigma = (I-\Theta)^{-1}(BB^T+\Psi)(I-\Theta)^{-T}$. The tuning parameter λ is determined by 5-fold cross validation. In both simulation studies, 50 replicates are carried out to draw summary statistics.

The performances of the proposed estimation method and algorithm are compared mainly under three cases, including (i) the latent factors are ignored, i.e. $K = 0$, which is equivalent to the method proposed by Shojaie and Michailidis (2010); (ii) the number of latent factors K is over-specified or under-specified, corresponding to overestimation or underestimation of the latent factors covariance W in a DAMG; and (iii) the number of latent factors K is selected by the proposed eigenvalue ratio (ER) method, i.e. $K = K_{ER}$. That is, $K_{\text{ER}} = \max_{K_{min}\leqslant k\leqslant K_{max}} \eta_k/\eta_{k+1}$, where η_k is the k^{th} largest eigenvalue of $\sum_{n=1}^{N}\mathbf{y}_n\mathbf{y}_n^T/(NP)$, with $K_{min} = 1$, $K_{max} = min(N,P)/2$.

For each simulated dataset, we generate the solution paths for the elements of Θ using a geometric sequence of values for λ, starting from the largest value $\lambda_{\max}$ at which $\widehat{\Theta}_{\lambda_{\max}} = \mathbf{0}$ and decreasing to the smallest value $\lambda_{\min} = 10^{-4}$. Note that the total number of detected edges increases as λ decreases. We then evaluate the performances of both estimation method and algorithm under different numbers of latent factors nested within a series of tuning parameter values. We also compare the performances of the proposed estimation method and algorithm with two top methods in the literature, namely the score-based method available in the R-package `sparsebn` (Aragam et al., 1956) (which is referred to as sparsebn hereafter),

and the PC-algorithm implemented in the R-package `pcalg` (Kalisch and Bühlmann, 2007), where the significance levels of the PC-algorithm are given by a geometric sequence of values $\{10^{-15}, \ldots, 0.95\}$. Neither the sparsebn nor the PC-algorithm requires the knowledge of node ordering as an input. An advantage of the SFEM is that it enables the use of partial knowledge on the node order to improve the statistical analysis. In practice, partial prior knowledge of biological network structure may be obtained from existing pathway databases, for example. To give the highest favor to these existing methods, when reporting the results from the software, we simply ignore the edge direction or equivalently assume the direction is always correctly detected.

4.2 *Findings from simulation studies*

Figure 2 shows two plots of the average number of correctly detected edges against the total number of detected edges over different numbers of latent factors K over 50 replicates. This figure appears in color in the electronic version of this article, and any mention of color refers to that version. Here "oracle" refers to the case where the proposed regularized estimation is carried out by using the true covariance matrix $W = BB^T + \Psi$ without estimating B and Ψ, namely the EM algorithm is not used in the estimation. We find that the proposed SFEM method in the case of $K = K_{ER}$ with estimated B and Ψ produces results very close to those obtained in the "oracle" case. This suggests that the EM algorithm works well to estimate the W matrix. Also, we see that the SFEM method with $K = K_{\text{true}}$, equal to 2 in the top panel (a) and 5 in the bottom panel (b) of Figure 2, outperforms all the other cases with misspecified K.

The performances of the PC-algorithm and the sparsebn method appear to be the worst, and are even worse than the SFEM with $K = 0$ where no latent factors are accounted for in the analysis. Figure 2 (a) shows that under a relatively light degree of masking ($K = 2$), the proposed SFEM($K = 0$) can gradually pick up more true signals when more false

discoveries are allowed. In contrast, neither the PC-algorithm nor the sparsebn method show any noticeable improvement. This is probably because both the PC-algorithm and the sparsebn method do not use any *a priori* knowledge of node ordering. In Figure 2(b) with 100 detected edges, the sparse SFEM with $K = K_{\text{ER}}$ can detect more than 95% of the true edges correctly with an average standard deviation of 1.45 edges, whereas the PC-algorithm or the score-based method can only detect about 10% of the true edges successfully. In other words, the unmeasured confounding factors can severely impair the performances of the PC-algorithm and the sparsebn method.

[Figure 2 about here.]

The quality of our method is further measured by the average number of true positive (TP), false positive (FP) and false negative (FN) edges, sensitivity (Sen) and Matthews correlation coefficient score (MCC). Table 1 summarizes the average performance of the SFEM with $K = K_{ER}$ for different numbers of P in the second simulation experiment with $K_{\text{true}} = 5$. For example, when $P = 200$, on average the estimated graph is able to identify 104.04 directed edges, of which 98.12 edges are the true edges, and the other 5.92 edges are false. In the case of $P = 200$, the number of parameters to be estimated is around $20,000$, which is much larger than the sample size $N = 100$. In this high-dimensional setting with a substantial amount of masking by $K = 5$ latent factors, results in Table 1 suggest that our regularization method can estimate the DAG structure with reasonable accuracy even with the limited sample size $N = 100$. When the network is relatively simpler with $P = 50$ or 100, the proposed estimation method and algorithm perform even better.

[Table 1 about here.]

Table 2 lists the results of both simulation experiments I and II with different numbers of latent factors and different percents of explained variability. Table 2 suggests that the proposed ER criterion works well in selecting the number of latent factors, except for the case

of Simulation II with PEV=1:2. This is because in this setting PEV is relatively small, and the ER criterion is always in favor of a stronger nuisance covariance structure with two latent factors. However, it is interesting to notice that, although the nuisance structure is slightly overestimated (i.e., one additional factor to the true $K = 1$), the resulting performance ($K_{ER} = 2$) still appears much better than that with an under-specified nuisance structure ($K = 0$), judging by, for example, MCC=0.85 versus 0.29. As shown in Table 2, either ignoring or under-specifying the number of latent factors results in abundant nonzero entries in Θ, many of which may be false edges. In contrast, if the number of factors is overestimated, the proposed method would produce a sparse Θ matrix, leading to many false negative discoveries. The latter presents a conservative analysis that fails to detect some of the true signals, which is often a more favorable scenario than the former, which reports excessive false signals. In summary, the proposed SFEM with K_{ER} shows a satisfactory performance with the highest sensitivity and MCC, as well as the lowest false discovery rate.

[Table 2 about here.]

Section 2 of the Supporting Information provides some additional simulation results for the comparison of SFEM, PC-algorithm, and sparsebn in both DAG simulation settings.

4.3 *Sensitivity analysis on the knowledge of node ordering*

An input of *a priori* node ordering presents a noticeable limitation on the proposed SFEM method. We further assess the performance of the proposed method under three scenarios: (i) fully known node ordering, (ii) fully unknown node ordering, and (iii) partially known node ordering. The second scenario is most likely to occur in practice, given that practitioners often know part of a network under investigation based on their own experiences and relevant publications. Here we use the setting of Simulation II with $P = 200$, $K = 5$, $N = 100$, $PEV = 1 : 4$, and $M = 100$. Figure 3 reports the results.

In scenario (ii) of fully unknown node ordering, we first apply the sparsebn method on

each of 50 simulated datasets $Y_{(s)}$, $s = 1, ..., 50$ to learn the underlying node ordering $\text{order}_{(s)}$ of the network. Reordering the nodes $Y_{(s)}$ based on the learned $\widehat{\text{order}}_{(s)}$ leads to a reordered $Y^*_{(s)}$. Finally, we apply our SFEM method on $Y^*_{(s)}$, $s = 1, ..., 50$. In scenario (iii) of partially known node ordering, our design is given as follows. Since the true DAG in the Simulation II design consists of 4 strong master regulators (or hubs), we randomly pick two of them and treat the corresponding sub-DAG as our prior knowledge about the network. So, we know *a priori* part of the true node ordering of the network, called $\text{order}_{(\text{prior})}$. For the rest of nodes, we once again learn the node ordering by the sparsebn method. We merge these two pieces as $(\widehat{\text{order}}^{\text{rest}}_{(s)}, \text{order}_{(\text{prior})})$ to form the node ordering of the network.

We also apply the proposed eigenvalue ratio method, which consistently selects $K = 5(100\%)$ under each scenario. Thus we compare the performance of our method $\text{SFEM}_{K=5}$ under the three levels of node ordering knowledge, as well as the naive PC-algorithm and sparsebn method that do not input any knowledge of node ordering. From Figure 3, with no surprise, our $\text{SFEM}_{K=5}$ method significantly outperforms the PC-algorithm and the sparsebn method in all scenarios. This figure appears in color in the electronic version of this article, and any mention of color refers to that version. Interestingly, accounting for latent factors with our SFEM method in scenario (ii) clearly helps boost the detection power compared to the sparsebn method that supplies the node ordering to the SFEM method. In the presence of such strong masking due to 5 unmeasured factors, it is certainly beneficial to use our SFEM method. Another important conclusion from this comparison is that knowing the node ordering partially can help a lot. The proposed SFEM method has the flexibility to accommodate some incomplete knowledge for improvement of detection power.

[Figure 3 about here.]

Under the same DAG setting, Section 2 of the Supporting Information provides an expanded simulation experiment II with 500 replicates. Section 3 of the Supporting Information

reports the results of average computation time for the EM-CD algorithm over different K values based on 50 rounds of simulations. It ranges from 3 minutes with $K = 0$ to 7 minutes with $K = 5$, which is reasonably fast given the size and complexity of the computational operations.

5. Analysis of METABRIC gene expression data

This section demonstrates the application of the proposed SFEM method to the METABRIC data, which consists of gene expression measurements collected from a study of the genomic landscape of breast cancers (Pereira et al., 2016). In the analysis of genetic regulatory networks, we focus on 82 driver genes identified by Pereira et al. (2016), which are measured from 1222 primary tumor samples. This set of driver genes is known for their individual causal effects on breast cancer outcomes, which have been established through somatic mutation patterns, which are independent of their gene expression profiles. Applying the proposed method, we hope to estimate DAGs involving these causal genes to learn about biological interactions and pathways relevant to the disease.

To obtain the node ordering required by our SFEM method, we first apply the sparsebn method to obtain an estimated ordering of 82 driver genes. We do not use the node ordering from the PC algorithm simply because it is sensitive to a pre-defined threshold required by the method. The SFEM method is then applied with K varying from 0 to 5. At K=0 (no latent factors), 211 edges are detected, some of which may be potentially masked by ubiquitous confounding in the experiment. When applying the eigenvalue ratio method to select K, we get $K = 2$, leading to 170 detected edges. The reduction of the detected edges seems to suggest that some of the detected edges at $K = 0$ can be explained by the unmeasured confounding that is accounted for with $K = 2$. Thus, the edges inferred at $K = 2$ are likely more robust. The related details can be found in the Supporting Information, Section 5.

To enhance the stability of the analysis results, we generated 50 bootstrap samples with

replacements from the gene expression data under the previously given node ordering. For each bootstrap sample, we apply the SFEM method in which K is determined by the eigenvalue ratio method, and the tuning parameter λ is selected by the 5-fold cross-validation. The final gene regulatory network is drawn following the majority voting strategy; that is, a final edge is reported only if it is detected at least 50% of the time out of 50 bootstrap samples. As shown in Table 5 of the Supporting Information, K=2 appears to be the dominant mode. In the final causal network voted by the 50 bootstrap samples, we detect 125 causal relationships among 71 genes. The detail of the regulatory network is shown in Figure 7 of the Supporting Information.

The gene regulatory network constructed by the SFEM shows some delicate structures among these breast cancer driver genes. Within the network, we find some interesting sub-networks, displayed in Figure 4. Unlike a star-shape topology where each driver gene independently causes the disease, our result reveals a pattern of complicated interactions between the driver genes. We find that the biggest hub is gene CCND2, which regulates the other 8 genes (BRCA1, JAK2, ABCC4, ERCC4, MLH1, DHRS13, LMO2, NFIB), while CCND2 is regulated by genes BIRC3 and FBN1. Another major hub is gene RUNX1, which regulates 7 genes (RAD51C, CCT2, BCL10, NDRG1, PTEN, HERPUD1, EXT1) while itself is regulated by TRIP11 and COL1A1. See Part A of Figure 4. In addition, we find that genes BRCA1 and LMO are two major offspring nodes, each of which is regulated by five genes. BRCA1 is a well-known breast cancer oncogene that is regulated by RAD51C, EZH2, RECQL4, NF1, and CCND2. Also, LMO is regulated by FH, PIK3C3, EZH2, CCND2, and FOXA1. See Part B of Figure 4. These intriguing results illustrate how pathway analysis can shed light on the regulatory mechanisms of these important disease genes.

[Figure 4 about here.]

Another real data example using cell signaling data is given in Section 4 of the Supporting

Information. This is a multivariate flow cytometry dataset that has been previously analyzed by many statisticians. Our analysis using the proposed SFEM method gives similar findings to those published.

6. Discussion

Given prior knowledge on node ordering among variables, we proposed a class of SFEMs for an exploratory analysis of causal network construction. The proposed methodology combines the structural equation model and the factor analysis model. Our SFEM method may be regarded as a general factor analysis model that enables to effectively segregate a DAG with directed edges from an acyclic directed mixed graph, where undirected edges induced by unmeasured confounding factors are identified and removed. In this way, a simpler and more interpretable causal network is obtained. When there are no latent factors included, the proposed SFEM reduces to the classical SEM. In this case, the reconstruction of DAGs based on our proposed L_1 norm regularization method is equivalent to the L_1 norm penalized likelihood method proposed by Shojaie and Michailidis (2010).

We developed a two-step EM-coordinate-descent algorithm for implementation of the proposed method that works reasonably well and can be applied to large networks, as shown in various numerical settings. However, our objective function for the whole set of parameters is non-convex, which might yield multiple local solutions in the optimization. Since both the CD algorithm and EM algorithm solve their respective convex functions, the algorithm convergence is certain. Finding a global optimal solution for non-convex problems is numerically very challenging, and worth further exploration. In addition, if information on node ordering is fully or partially unavailable, our method can incorporate an estimated ordering obtained from existing methods (e.g., the score-based method). Our simulation studies have demonstrated a clear improvement of the proposed SFEM method on detection power over existing methods in the presence of masking factors. We expect that our method

can further improve detection power given better estimation of causality direction among network nodes. In addition, in the real data analysis, causal relations in gene regulatory networks are possibly nonlinear and may not be detectable using the linear SFEM proposed in this paper. Learning nonlinear causality presents another interesting extension of this research topic.

Acknowledgments

The authors are grateful to Drs. Ji Zhu and Matthias Kretzler for their constructive comments on an early draft of this paper. They like to thank the Co-Editor, an associate editor, and two referees, for their valuable suggestions that significantly improved the clarity of the manuscript. Song's research is supported by an NIH grant (R01ES024732) and an NSF grant (DMS1181734).

References

Ahn, S. C. and Horenstein, A. R. (2013). Eigenvalue ratio test for the number of factors. *Econometrica* **81,** 1203–1227.

Anandkumar, A., Hsu, D., Javanmard, A., and Kakade, S. (2013). Learning linear Bayesian networks with latent variables. In *Proceedings of the 30th International Conference on Machine Learning*, pages 249–257.

Aragam, B., Gu, J., and Zhou, Q. (1956). Learning large-scale Bayesian networks with the sparsebn package. *Journal of Statistical Software* **91,** 1–38.

Aragam, B. and Zhou, Q. (2015). Concave penalized estimation of sparse Gaussian Bayesian networks. *Journal of Machine Learning Research* **16,** 2273–2328.

Bai, J. and Li, K. (2012). Statistical analysis of factor models of high dimension. *Annals of Statistics* **40,** 436–465.

Bai, J. and Ng, S. (2002). Determining the number of factors in approximate factor models. *Econometrica* **70,** 191–221.

Bickel, P. J. and Levina, E. (2008). Covariance regularization by thresholding. *Annals of Statistics* **36,** 2577–2604.

Caner, M. and Han, X. (2014). Selecting the correct number of factors in approximate factor models: The large panel case with group bridge estimators. *Journal of Business and Economics Statistics* **32,** 359–374.

Cheng, J., Greiner, R., Kelly, J., Bell, D., and Liu, W. R. (2002). Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence* **137,** 43–90.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics* **32,** 407–451.

Ellis, B. and Wong, W. H. (2008). Learning causal Bayesian network structures from experimental data. *Journal of the American Statistical Association* **103,** 778–789.

Friedman, N. and Koller, D. (2003). Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning* **50,** 95–125.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of Computational Biology* **7,** 601–620.

Fu, F. and Zhou, Q. (2013). Learning sparse causal Gaussian networks with experimental intervention: regularization and coordinate descent. *Journal of the American Statistical Association* **108,** 288–300.

Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., and Young, R. A. (2002). Combining location and expression data for principled discovery of genetic regulatory network models. In *Pacific Symposium on Biocomputing*, volume 7, pages 437–449.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks - The combination of knowledge and statistical data. *Machine Learning* **20,** 197–243.

Hirose, K. and Konishi, S. (2012). Variable selection via the weighted group lasso for factor analysis models. *Canadian Journal of Statistics* **40,** 345–361.

Huang, J., Liu, N., Pourahmadi, M., and Liu, L. (2006). Covariance matrix selection and estimation via penalised normal likelihood. *Biometrika* **93,** 85–98.

Johnson, R. A. and Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis*. Pearson Prentice Hall, 6th edition.

Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs

with the pc-algorithm. *Journal of Machine Learning Research* **8,** 613–636.

Lam, W. and Bacchus, F. (1994). Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence* **10,** 269–293.

Lawley, D. N. and Maxwell, A. E. (1962). Factor analysis as a statistical method. *Journal of the Royal Statistical Society Series D* **12,** 209–229.

Levina, E., Rothman, A., and Zhu, J. (2008). Sparse estimation of large covariance matrices via a nested lasso penalty. *The Annals of Applied Statistics* **2,** 245–263.

Li, F. and Yang, Y. (2004). Recovering genetic regulatory networks from micro-array data and location analysis data. *Genome Informatics Series* **15,** 131.

Li, F. and Yang, Y. (2005). Using modified lasso regression to learn large undirected graphs in a probabilistic framework. In *Proceedings of the 20th National Conference on Artificial Intelligence*, volume 20, pages 801–806. AAAI Press.

Lopes, C. T., Franz, M., Kazi, F., Donaldson, S. L., Morris, Q., and Bader, G. D. (2010). Cytoscape web: an interactive web-based network browser. *Bioinformatics* **26,** 2347–2348.

Onatski, A. (2009). Testing hypotheses about the number of factors in large factor models. *Econometrica* **77,** 1447–1479.

Onatski, A. (2010). Determining the number of factors from empirical distribution of eigenvalues. *The Review of Economics and Statistics* **92,** 1004–1016.

Pe'er, D., Regev, A., Elidan, G., and Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics* **17,** S215–S224.

Peng, J., Zhu, J., Bergamaschi, A., Han, W., Noh, D. Y., Pollack, J. R., and Wang, P. (2010). Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *Annals of Applied Statistics* **4,** 53–77.

Pereira, B., Chin, S. F., Rueda, O. M., Vollan, H. K., Provenzano, E., Bardwell, H. A., and et al. (2016). The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature Communications* **7:11479,**.

Pourahmadi, M. (1999). Joint mean-covariance models with applications to longitudinal data: Unconstrained parameterisation. *Biometrika* **86,** 677–690.

Robinson, R. W. (1973). *Counting Labeled Acyclic Digraphs.* Academic Press, New York.

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N. (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics* **34,** 166–176.

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

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

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning* **65,** 31–78.

Wu, W. B. and Pourahmadi, M. (2003). Nonparametric estimation of large covariance matrices of longitudinal data. *Biometrika* **90,** 831–844.

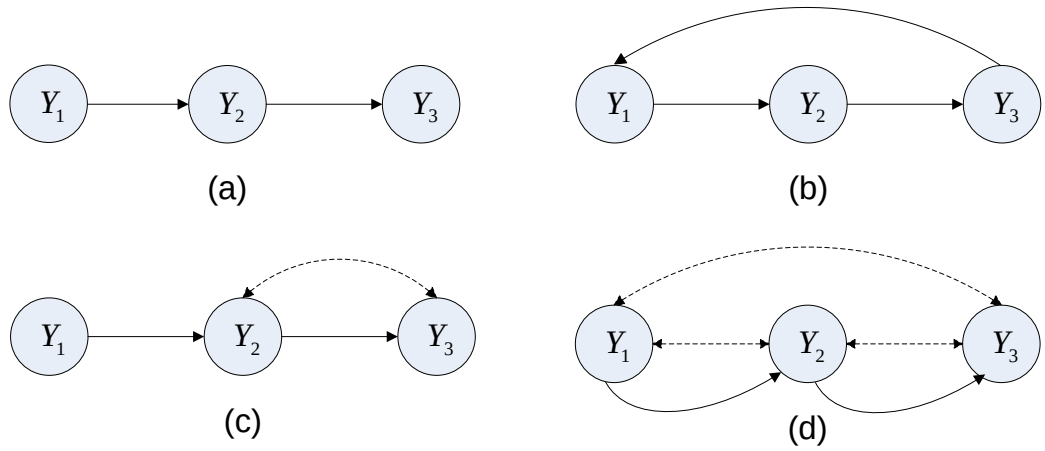
Zhou, Q. (2011). Multi-domain sampling with applications to structural inference of Bayesian networks. *Journal of the American Statistical Association* **106,** 1317–1330.

SUPPORTING INFORMATION

Web Appendices, Tables, and Figures referenced in Sections 4 and 5 are available with this paper at the Biometrics website on Wiley Online Library. In addition, some of the computing code used in the simulation studies is available in Section 6 of the Supporting Information.

Figure 1: Part A presents four examples of directed mixed graphs. The graph in A(b) is cyclic, while all others are acyclic. An arrowed solid line indicates a directed edge and a dashed line denotes an undirected (or bi-directed) edge. Part B presents an acyclic directed mixed graph that contains a DAG with the directed edges (arrowed solid lines) among nine observed variables $y_1, \ldots, y_9$ and a set of undirected edges induced by 3 common latent factors z_1, z_2 and z_3. This figure appears in color in the electronic version of this article, and any mention of color refers to that version.

A. Four examples of directed mixed graphs.



B. An acyclic directed mixed graph containing a DAG.

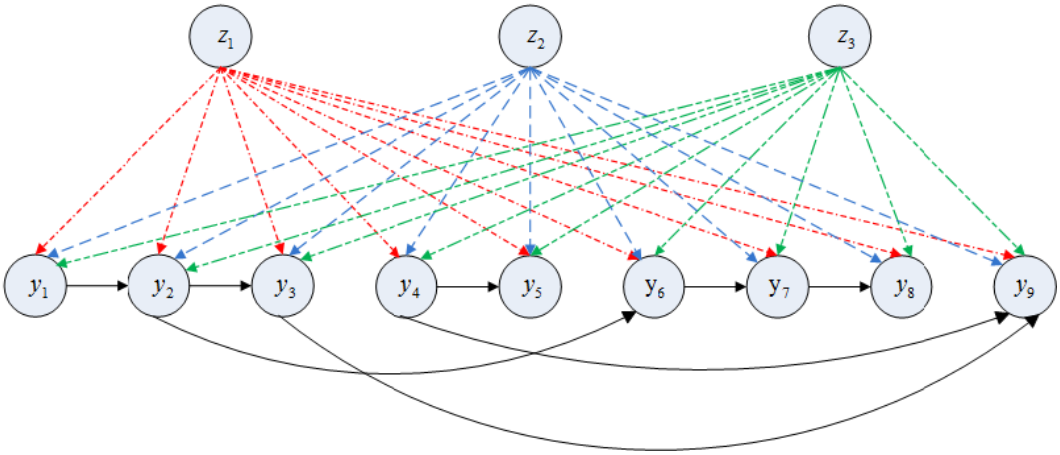
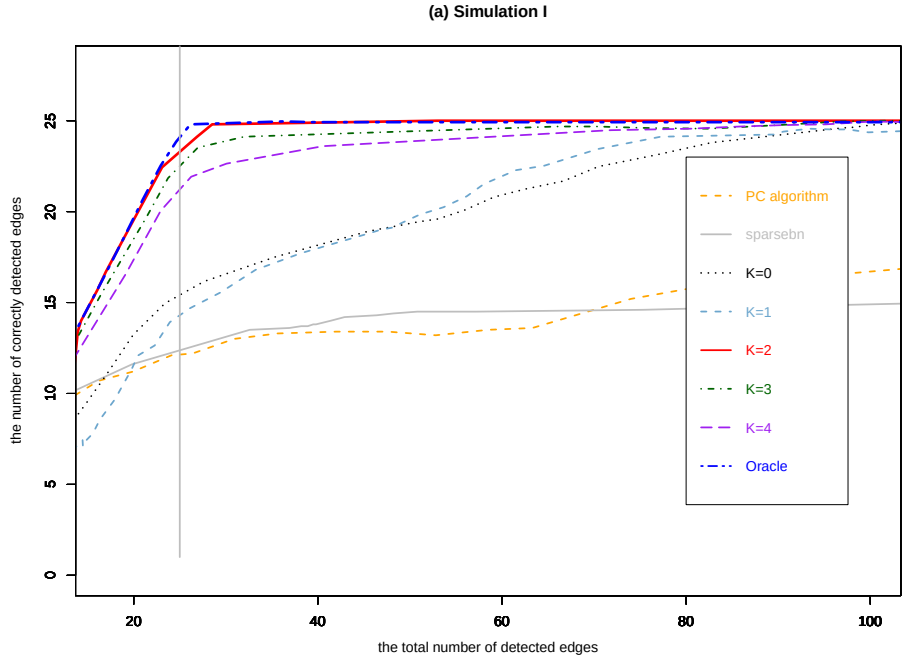


Figure 2: Summary results from two DAGs designed in the simulation studies. The x-axis is the total number of detected edges, and the y-axis is the average number of correctly identified edges over 50 replicates. The vertical (gray) line corresponds to the number of true edges. Panel (a) displays the results from the first small DAG simulation design with $P = 50, M = 25, K = 2, N = 25$ as well as the estimated $K_{ER} = 2$. Panel (b) shows the results of the second large DAG simulation design with $P = 200, M = 100, K = 5, N = 100$ as well as the estimated $K_{ER} = 5$. This figure appears in color in the electronic version of this article, and any mention of color refers to that version.



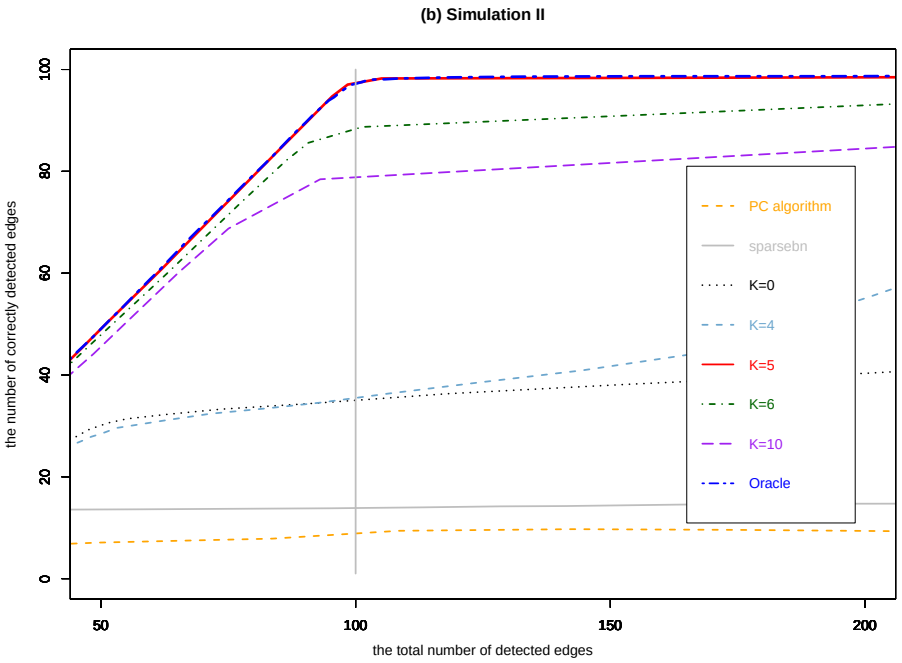


Figure 3: Summary results based on the large DAG simulation design with $P = 200, M = 100, K = 5, N = 100$ as well as the estimated $K_{ER} = 5$. The x-axis is the total number of detected edges, and the y-axis is the number of correctly identified edges averaged over 50 replicates. The vertical (black) line corresponds to the number of true edges. This figure appears in color in the electronic version of this article, and any mention of color refers to that version.

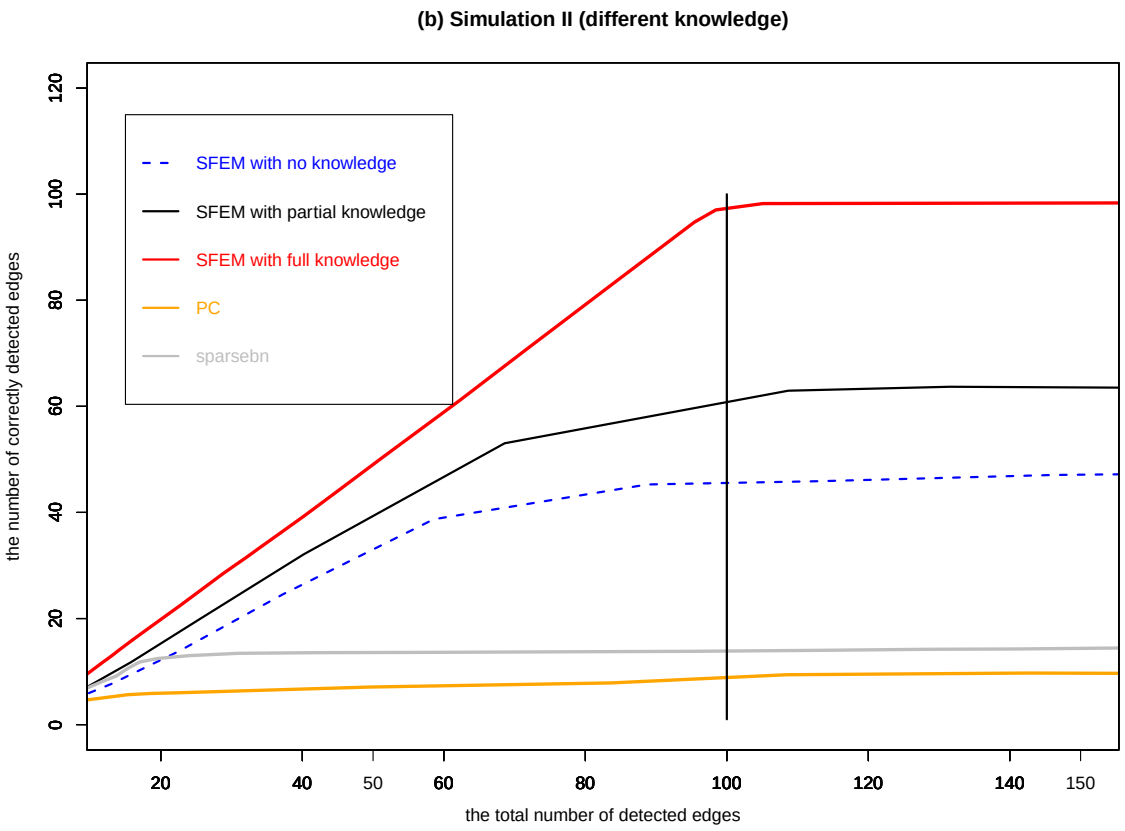
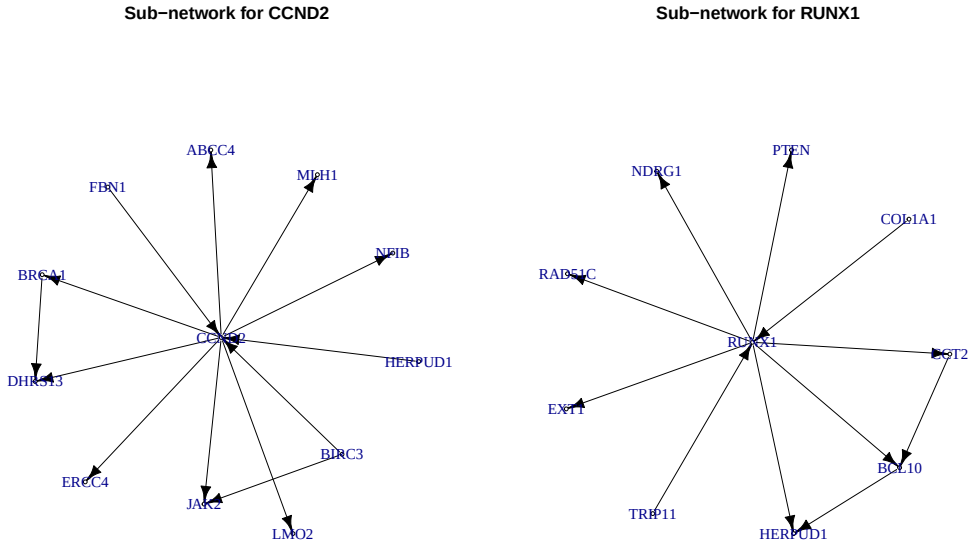


Figure 4: Sub-networks for master regulator genes (CCND2 and RUNX1) and master offspring genes (BRCA1 and LMO), respectively. This figure appears in color in the electronic version of this article, and any mention of color refers to that version.

A. Sub-networks for master regulator genes.



B. Sub-networks for master offspring genes.

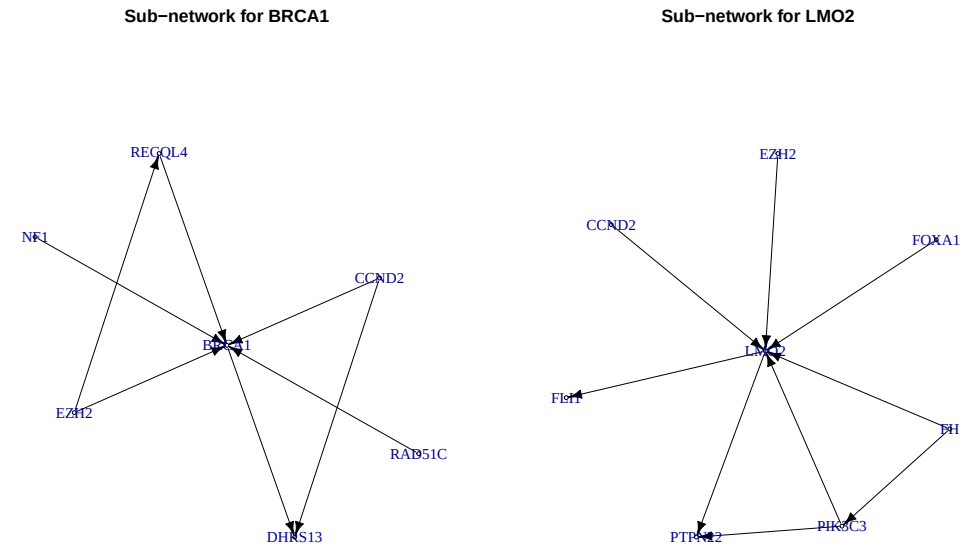


Table 1: Performance comparison under different number of nodes, $P = 50, 100, 200$ under the second large DAG simulation design with $M = 100, K = 5$ and $N = 100$.

P	Total (TP+FP)	TP	FP	FN	Sen	MCC	$K_{ER}(\%)$
50	99.07	97.12	1.95	1.94	0.98	0.99	5 (100%)
100	100.69	97.56	3.13	2.17	0.97	0.97	5 (100%)
200	104.04	98.12	5.92	1.88	0.98	0.96	5 (100%)

Table 2: Results from both small and large DAG simulation designs, respectively, where the number of latent factors K and the percent of explained variability (PEV) varies over four cases.

PEV	K_{true}	Method	Total (TP+FP)	TP	FP	FN	Sen	MCC	K_{ER}(%)
Simulation I									
1:3	2	SFEM_{ER}	29.44	24.76	4.68	0.24	0.99	0.92	2 (100%)
		$\text{SFEM}_{K=0}$	118.84	24.96	93.88	0.04	0.99	0.44	
		$\text{SFEM}_{K=5}$	25.88	20.24	5.64	4.76	0.81	0.80	
Simulation II									
1:4	5	SFEM_{ER}	104.04	98.12	5.92	1.88	0.98	0.96	5 (100%)
		$\text{SFEM}_{K=0}$	1530.92	97.96	1432.96	2.04	0.98	0.24	
		$\text{SFEM}_{K=7}$	72.29	63.86	8.43	36.14	0.64	0.70	
1:2	1	SFEM_{ER}	88.2	79.84	8.34	20.16	0.80	0.85	2(100%)
		$\text{SFEM}_{K=1}$	104.44	97.44	7.00	2.56	0.97	0.95	
		$\text{SFEM}_{K=0}$	1015.32	93.64	921.68	6.36	0.94	0.29	
1:6	10	SFEM_{ER}	93.76	91.52	2.24	8.48	0.92	0.94	10 (100%)
		$\text{SFEM}_{K=0}$	3686.72	97.64	3589.08	2.36	0.98	0.14	
		$\text{SFEM}_{K=15}$	53.28	49.64	3.64	50.36	0.50	0.67	