HHS Public Access

Author manuscript

J Am Stat Assoc. Author manuscript; available in PMC 2017 October 18.

Published in final edited form as:

J Am Stat Assoc. 2016 ; 111(515): 1004–1019. doi:10.1080/01621459.2016.1142880.

Estimation of Directed Acyclic Graphs Through Two-stage Adaptive Lasso for Gene Network Inference

Sung Won Han[1], **Gong Chen**[2], **Myun-Seok Cheon**[3], and **Hua Zhong**[1,*]

[1]Division of Biostatistics, Departments of Population Health, New York University, New York, NY, USA, 10016

[2]Pharmaceutical Sciences, Pharma Early Research and Development, Roche Innovation Center New York, New York, NY, USA

[3]School of Industrial and System Engineering, Georgia Institute of Technology, Atlanta, GA, USA, 30332

Abstract

Graphical models are a popular approach to find dependence and conditional independence relationships between gene expressions. Directed acyclic graphs (DAGs) are a special class of directed graphical models, where all the edges are directed edges and contain no directed cycles. The DAGs are well known models for discovering causal relationships between genes in gene regulatory networks. However, estimating DAGs without assuming known ordering is challenging due to high dimensionality, the acyclic constraints, and the presence of equivalence class from observational data. To overcome these challenges, we propose a two-stage adaptive Lasso approach, called NS-DIST, which performs neighborhood selection (NS) in stage 1, and then estimates DAGs by the Discrete Improving Search with Tabu (DIST) algorithm within the selected neighborhood. Simulation studies are presented to demonstrate the effectiveness of the method and its computational efficiency. Two real data examples are used to demonstrate the practical usage of our method for gene regulatory network inference.

Keywords

Directed acyclic graphs; Lasso estimation; Neighborhood selection; Probabilistic graphical model; Structure equation model

1 INTRODUCTION

The genome encodes thousands of genes, the messenger RNA (mRNA) of which are related to numerous cellular functions such as cell survival. Fundamentally, regulatory processes are carried out by regulatory genes, called transcription factors (TFs), which activate, repress, or modulate the transcription of their downstream genes (Materna and Oliveri, 2008). All regulatory genes are themselves under the control of transcriptional regulators and together form large gene regulatory networks (GRNs) (Markowetz and Spang, 2007; Materna and

*Corresponding author. judy.zhong@nyumc.org.

Oliveri, 2008). Graphical models are a popular approach to find dependence and conditional independence relationships between multiple interacting quantities, i.e., expression levels of different genes in the GRNs (Friedman et al., 2000; Imoto et al., 2003; Beal et al., 2005; Zou, 2006; Werhli et al., 2006). This method is also widely used in biological applications such as in gene-gene expression networks (Friedman et al., 2000; Butte et al., 2000; Oldham et al., 2006; Keller et al., 2008), protein-protein interaction analysis (Jansen et al., 2003; Tu et al., 2012), phenotype networks (Neto et al., 2008, 2010), causal networks linking gene expression and metabolic change (Ferrara et al., 2008), and metabolomics (Steuer, 2006).

The estimation of directed gene networks has been very challenging since the number of genes is very large, measuring in the tens of thousands, while sample sizes are small, mostly a few hundred. In addition, gene networks have many regulatory genes (i.e., TFs), so they have a structure with several hub nodes, which causes multicollinearity problems around the hub nodes. Such challenges naturally lead to two related problems: variable selection in high dimension and estimation of directed graphical models.

For the variable selection problem, BIC criteria, which is a special type of L_0-penalty, is widely suggested as the best subset selection technique (Shao, 1997; Shi and Tsai, 2002). However, the best subset selection is not computationally efficient in high dimensional settings, and the selected subset is variable due to the discreteness of L_0 penalty (Breiman, 1995; Fan and Li, 2001). To address this, Tibshirani (1996) proposed the Lasso, which gives a more stable estimate than the subset variable selection method, and has been popularly used for simultaneous estimation and variable selection. Recently, Zou (2006) proposed the adaptive lasso, and showed that it provides consistent variable selection under a certain selection of weights.

Directed acyclic graphs (DAGs) are a special class of directed graphical models, where all the edges are directed edges and contain no directed cycles (Pearl, 2000). A connection in DAGs can be made between their characterization and a causal inference by applying the directed Markov property under certain assumptions (Pearl, 2000; Friedman et al., 2008). The skeleton of a DAG, obtained by ignoring the directions of a DAG, is in general different from the undirected graphical models (Lauritzen, 1996). The two nodes are connected in the skeleton of a DAG if and only if they are dependent given any subset of all the other nodes (Lauritzen, 1996). Therefore, the skeleton of the DAG is typically a subset of the conditional independence graph, which includes additional edges (Lauritzen, 1996; Kalisch and Buhlmann, 2007; Shojaie and Michailidis, 2010).

Estimating the structure of DAGs has been the subject of extensive studies during the last decade (Buntine, 1994, 1996; Heckerman et al., 1995; Neapolitan, 2004; Daly et al., 2011), and it is especially challenging in high dimensional settings as the number of DAGs grows super exponentially in the number of nodes (Robinson, 1977). Three major types of approaches have been developed for estimating the structures: (1) a score-and-search approach through the structure space, (2) a constraint-based approach that tests conditional independence identified in the data, and (3) a hybrid approach that combines both the score-and-search approach and the constraint-based approach. The score-and-search approach (Broom and Subramanian, 2006; Buntine, 1994, 1996; Heckerman et al., 1995; Neapolitan,

2004) searches for a DAG by maximizing a score function, which often consists of a model fitting part and a penalty of model complexity (Daly et al., 2011). Heuristic or ad hoc search algorithms are often employed to search for a high-scoring DAG without enumerating all possible structures. The constraint-based approach uses statistical tests to estimate whether certain conditional independence between the variables hold. One popular example is the PC-algorithm designed by Spirtes et al. (2000). Ramsey (2006) developed a local adaptation of the PC-algorithm, called Markov Blanket Fan Search, which uses conditional independence tests over causally sufficient variables in high dimensional data. Kalisch and Bühlmann (2007) showed that the PC-algorithm has uniform consistency for sparse high dimensional DAGs. Kalisch et al. (2012) implemented the PC-algorithm efficiently for high dimensional sparse networks (Kalisch et al., 2010; Nagarajan et al., 2010; Stekhoven et al., 2012). Recently, Colombo and Maathuis (2013) developed a modification of the PC-algorithm, called the PC-stable algorithm, that removes part or all of the order dependence. The hybrid approach is especially computationally efficient in high dimension since it divides the solution search procedure into two-stages, where the first stage finds the undirected graph or skeleton, and the second stage estimates the directionality. For example, the max-min hill-climbing (MMHC) algorithm (Tsamardinos et al., 2006) estimates the skeleton by the technique from a constraint-based approach, and then finds the directionality by a score-and-search approach. Neto et al. (2008) built an undirected graph, then compared the likelihood ratios corresponding to the two directions for each edge to estimate a DAG. Pellet and Elisseeff (2008) applied feature selection algorithms (John et al., 1994; Guyon and Elisseeff, 2003) to estimate a Markov blanket or skeleton, then decided directionality by utilizing the v-structure patterns. Recently, Ha et al. (2015) proposed a two-stage approach, called PenPC, which estimates an undirected graph by a penalized regression, and removes false connections by applying a modified PC-stable algorithm, which finally leads to an estimate of a complete partial DAG.

For the score-and-search approach, several methods have been proposed to estimate the structure of graphical models through penalized likelihood, with most efforts focused on undirected graphs with L_1-penalty (Yuan and Lin, 2007; Friedman et al., 2008; Meinshausen and Buhlmann, 2006). Recently, penalized likelihood approaches have been developed to estimate DAGs. Yuan et al. (2012) estimated multiple DAGs based on the L_0-penalized likelihood with known variable orders and known variance of latent variables. Chickering (2002b) proposed a greedy equivalence search (GES) algorithm, which incorporates stepwise (forward and backward) search with equivalence space search under the L_0-penalty. He argued that GES provides the optimal solution as it considers all possible combinations of parents' nodes given each child node. Schmidt et al. (2007) estimated an undirected graph based on the L_1-penalized linear regression and applied a permutation technique to variable orders to estimate a DAG. Since the entire permutation imposes a heavy computational burden, they proposed an ad hoc approach that swaps adjacent ordered variables. In addition, Raskutti and Uhler (2013) proposed the sparsest permutation algorithm based on finding the permutation of the variables that yields the sparsest DAG.

In this paper, we propose the two-stage adaptive lasso regression approach as the score-and-search approach to estimate DAGs in high dimension under unknown variable ordering. Meinshausen and Buhlmann (2006) proved that variable selection by lasso regression leads

to finding (probabilistic) neighbors, which indicate probabilistic dependency between two variables given others. In addition, Zou (2006) proposed the adaptive lasso regression, which gives asymptotic consistency in variable selection. Fu and Zhou (2013) proposed a profiled likelihood with an adaptive lasso penalty to estimate DAGs under unknown variable order based on experimental data with interventions, and used a blockwise coordinate descent (CD) algorithm to find a local optimal solution. Aragam and Zhou (2015) improved the CD algorithm under the observational data. Shojaie and Michailidis (2010) proposed a likelihood with an adaptive lasso penalty to estimate DAGs under known variable order and equal variance of latent variables. The known ordering of variables was exploited to reformulate the likelihood as a function of the coefficient matrix of the graph. Here, we propose a two stage adaptive lasso approach, where the first stage selects probabilistic neighborhood, and the second stage estimates the DAG within the identified probabilistic neighborhood. In Stage 2, we propose an efficient searching algorithm based on discrete improving search with steepest descent, cycle elimination, and tabu list.

The structure of this paper is as follows: Section 2 discusses the formulation of the model and the score function based on the adaptive Lasso framework, and Section 3 describes the two stage metaheuristic algorithm. In Section 4, we present results of a simulation study, and, in Section 5, we apply our method to two real applications. Finally, we conclude our results in Section 6.

2 PROBLEM FORMULATION

In this section, we discuss the graphical representation of variables and an adaptive lasso frame-work.

2.1 Graphical Representation

Each gene's expression corresponds to a random variable, and a causal effect between two genes corresponds to the causal relationship between the corresponding variables. Suppose that we have p variables, $X_1, X_2, \ldots, X_p$, and the number of observations is n. Denote the $n \times p$ data matrix by χ. The variable and causal relation can be represented by a node and directed edge in a graph $G = (V, E)$, where V indicates the set of p nodes and $E(\subset V \times V)$ indicates edge sets. We assume that (j, i) is not in E if (i, j) belongs to E. We denote j as a parent given an edge $j \rightarrow i$, and the set of parent nodes for node i is denoted by pa_i. We define $\mathbf{T}$ as the network structure of the $p \times p$ matrix whose (i, j) entry is 1 given the presence of the edge $j \rightarrow i$. $\mathbf{T}$ is simply the transposition of an adjacency matrix in the Graph Theory (West, 2001). We assume there is no cycle in the graph. The following lemma establishes the relationship between T and the acyclic assumption.

Lemma 2.1—*The network structure* $\mathbf{T}$ *is acyclic if and only if*

$$\sum_{l=1}^{\min(\mathrm{Card}(\boldsymbol{T}),p)} \sum_{m=1}^{p} \left[\boldsymbol{T}^l\right]_{m,m} = 0, \quad (1)$$

where Card($\mathbf{T}$) *is the number of nonzero entries, which indicates a cardinality of* $\mathbf{T}$.

The proof is trivial from the property of a directed cycle in terms of the adjacency matrix (West, 2001) or from the property of a nonrecurrent state by treating **T** as a transition matrix in a discrete Markov chain (Ross, 1996).

2.2 Linear Structural Equation Model

We can represent causal relationship in a graph in the following linear regression form (Pearl, 2000):

$$X_i = \sum_{j \in pa_i} a_{ij} X_j + Z_i, \quad (2)$$

where a_{ij} is a coefficient indicating a directional effect from a parent j to a child i, and Z_i is a latent variable, which is not observed and indicates an unexplained variation. We assume that Z_is follow independent normal distributions with mean 0 and variance σ_j^2. a_{ij} indicates a partial covariance between X_i and X_j given other parents of X_i. If the observations of X_is are standardized, a_{ij} indicates a partial correlation. The representation by the matrix notation is as follows. Let $\mathbf{X} = [X_1, X_2, \ldots, X_p]^T$, and $\mathbf{Z} = [Z_1, Z_2, \ldots, Z_p]^T \sim MN(0, \mathbf{D})$, where **D** is a diagonal matrix with elements of $\sigma^2_{Z_1}, \sigma^2_{Z_2}, \ldots, \sigma^2_{Z_p}$. Then, we can represent the set of all coefficients, a_{ij}s, shown in Equation (2) by the coefficient matrix **A**:

$$\mathbf{A} = \begin{pmatrix} 0 & a_{X_1X_2} & \cdots & a_{X_1X_{p-1}} & a_{X_1X_p} \\ a_{X_2X_1} & 0 & \cdots & a_{X_2X_{p-1}} & a_{X_2X_p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{X_{p-1}X_1} & a_{X_{p-1}X_2} & \cdots & 0 & a_{X_{p-1}X_p} \\ a_{X_pX_1} & a_{X_pX_2} & \cdots & a_{X_pX_{p-1}} & 0 \end{pmatrix}.$$

Thus, Equation (2) can be rewritten in the form of $\mathbf{X} = \mathbf{AX} + \mathbf{Z}$, where a_{ij} is the (i, j) entry of **A**. Then, the compact form becomes $\mathbf{X} = \Lambda\mathbf{Z}$, where $\Lambda = (I - \mathbf{A})^{-1}$. Correspondingly, **X** follows a multivariate normal distribution with $E[\mathbf{X}] = \Lambda\mu_{\mathbf{Z}}$ and $Var[\mathbf{X}] = \Lambda\mathbf{D}\Lambda^T$. Therefore, we can estimate a DAG G by estimating the matrix **A**, while the variance matrix of the latent variables **D** is the nuisance parameter. For notational convenience, we define $\mathbf{T}^{\mathbf{A}}$ as $\mathbf{T}^{\mathbf{A}}_{ij} = 1$ if $A_{ij} \neq 0$. $\mathbf{T}^{\mathbf{A}}$ is therefore used to reflect the acyclic restriction by Lemma 2.1 for estimating **A**. To regularize the scale of the coefficients, we standardize the data by centering and scaling such that for the i_{th} variable, $\bar{x}_i = 0$ and $\sum_{k=1}^{n} (x_{ik} - \bar{x}_i)^2/n = 1$. Unless otherwise Specified, the data are assumed to be standardized.

2.3 Score Function Formulation

The log likelihood presented by matrix **A** in Equation (3) is

$$-\frac{2}{n}\log\prod_{i=1}^{n} f(\mathbf{X}_{(i)}) \propto -\log\,|(\mathbf{I}-\mathbf{A})^T\mathbf{D}^{-1}(\mathbf{I}-\mathbf{A})| + \mathrm{tr}\,\left[\mathbf{D}^{-1}(\mathbf{I}-\mathbf{A})\mathbf{S}(\mathbf{I}-\mathbf{A})^T\right], \quad (3)$$

where $\mathbf{X}_{(i)}$ is the i_{th} observation of $\mathbf{X}$, $\mathbf{I}$ is a $p \times p$ identity matrix, and $\mathbf{S}$ is a sample covariance matrix defined by $\mathbf{S}=\sum_{i=1}^{n}(\mathbf{X}_{(i)}-\overline{X})(\mathbf{X}_{(i)}-\overline{X})^{T}/n$ with $\overline{X}=\sum_{i=1}^{n}\mathbf{X}_{(i)}/n$. Due to the acyclic assumption, $\log|\mathbf{I}-\mathbf{A}|=0$. We assume that a graph is sparse with respect to the number of edges. To impose the sparsity constraint, we adopt a penalty function for the likelihood in Equation (3). The penalized log likelihood after simplification can be written as follows:

$$\sum_{i=1}^{p}\left[\log\left(\sigma^{2}_{Z_k})+\frac{1}{\sigma^{2}_{Z_k}}\left[(\mathbf{I}-\mathbf{A})\mathbf{S}(\mathbf{I}-\mathbf{A})^{T}\right]_{k,k}\right]+\lambda\mathbf{J}(\mathbf{A}).\right.$$

In this paper, we consider an adaptive lasso with $\mathbf{J}(\mathbf{A}) = \sum_{i,j=1:p,i\neq j} w_{ij}|A_{ij}|$. With $\sigma^{2}_{Z_i}$ replaced by the residual sum of the square $[(\mathbf{I}-\mathbf{A})\mathbf{S}(\mathbf{I}-\mathbf{A})^{T}]_{k,k}$, the penalized likelihood function of $\mathbf{A}$ becomes a profiled likelihood such as

$$\sum_{k=1}^{p}\log\left[(\mathbf{I}-\mathbf{A})\mathbf{S}(\mathbf{I}-\mathbf{A})^{T}\right]_{k,k}+\lambda\mathbf{J}(\mathbf{A}). \quad (4)$$

The first order Taylor expansion of Equation (4) becomes

$$\sum_{i=1}^{p}\left[(\mathbf{I}-\mathbf{A})\mathbf{S}(\mathbf{I}-\mathbf{A})^{T}\right]_{k,k}+\lambda\mathbf{J}(\mathbf{A}), \quad (5)$$

which is equivalent to the penalized likelihood under the equal variance assumption of latent variables in Shojaie and Michailidis (2010). The final form of Equation (5) is

$$S(\mathbf{A})-\sum_{k=1}^{p}\left[\frac{1}{n}\|\chi_k-\chi\mathbf{a}_k\|_2^2+\lambda\sum_{j=1}^{p}w_{kj}|a_{kj}|\right], \quad (6)$$

where $\mathbf{T}^{\mathbf{A}}$ satisfies Equation (1). In the score function, χ is an $n \times p$ data matrix, and χ_k is a data vector at k_{th} variable. $\mathbf{a}_k$ is a coefficient vector, $[a_{k1}, a_{k2}, \ldots, a_{k(p-1)}, a_{kp}]^T$, which is a column vector representing a k_{th} row in matrix $\mathbf{A}$.

The score function (6) has several convenient properties. For one, the approximated score function (6) is convex under the lasso penalty, although the acyclic restriction is nonconvex. Furthermore, the score function (6) is a row separable Lasso function if we ignore the acyclic constraint. These features allow one to use more efficient algorithms to optimize score function (6).

To decide the weight, w_{ij}, Zou (2006) proposed $w_{ij}=\frac{1}{|\tilde{A}_{ij}|^{\gamma}}$ for some power $\gamma > 0$, where $\tilde{A}_{ij}$ is a certain initial estimate. Zou (2006) showed that the adaptive lasso satisfies the

consistency in model selection if $\tilde{A}_{ij}$ is a $\sqrt{n}$-consistent estimate of A_{ij} and suggested using the ordinary least squares (OLS) estimate for $\tilde{A}_{ij}$. Fu and Zhou (2013) proposed the ordinary least squares (OLS) estimate with an upper bound, which is $(\frac{1}{|\tilde{A}_{ij}|^{\gamma}}, \frac{1}{(N^{-1})^{\gamma}})$ with $N = 10^4$. However, if there are correlations among variables, the estimates from OLS are unstable. For alternative approaches, Shojaie and Michailidis (2010) proposed the initial weights by using

$$w_{ij}=\max(1, \frac{1}{|\tilde{A}_{ij}|^{\gamma}}). \quad (7)$$

where $\tilde{A}_{ij}$ is estimated from the regular lasso regression with initial penalty parameter $\lambda_0 > 0$. Shojaie and Michailidis (2010) showed that the approach with $\tilde{A}_{ij}$ estimated from the regular lasso regression shows better performance than that based on the ordinary least squares. λ_0 can be chosen to be the same as λ, but it is recommended to use a smaller value of λ_0 to avoid an over-sparse solution (Shojaie and Michailidis, 2010).

To assign an upperbound to w_{ij} in formula (7), we propose

$$w_{ij}=\max\left(1, \min(\frac{1}{|\tilde{A}_{ij}|^{\gamma}}, \frac{1}{(N^{-1})^{\gamma}})\right), \quad (8)$$

where $\tilde{A}_{ij}$ is estimated from the regular lasso regression with initial penalty parameter $\lambda_0 > 0$. The formula (8) provides the lower and upper bound of 1 and N^{γ}, respectively. In our simulation study, we use $N = 10^4$ as Fu and Zhou (2013) did. We construct the initial estimates $\tilde{A}_{ij}$ from the regular lasso estimates by minimizing function (8) with a certain λ_0, γ, and $w_{ij} = 1$. Based on the simulation study, we suggested λ_0=0.1 and $\gamma = 0.15$ (Section 4 of Supplementary Materials).

3 TWO STAGE SOLUTION SEARCH ALGORITHM

In this section, we discuss a solution search algorithm to minimize the proposed score function in Equation (6). Due to the combinatorial complexity from the acyclic constraint, the optimization problem of function (6) cannot be directly transformed into an equivalent penalized regression problem, which complicates the minimization problem. Therefore, we propose a two stage solution search algorithm, called Neighborhood Selection, followed by Discrete Improving Search with tabu list (NS-DIST), that provides a high quality solution with a reasonable computational time.

3.1 Stage 1: Neighborhood Selection (NS)

In Stage 1, we estimate the conditional independence graph by the probabilistic neighborhood selection approach proposed by Meinshausen and Buhlmann (2006). In our work, this is to minimize score function

$$\sum_{k=1}^{p}\left[\frac{1}{n}\|\chi_k-\chi\beta_k\|_2^2+\lambda\sum_{j=1}^{p}w_{kj}|\beta_{kj}|\right], \quad (9)$$

where β_k is a coefficient vector, $[\beta_{k1}, \beta_{k2}, \ldots, \beta_{k(p-1)}, \beta_{kp}]^T$, which is a column vector representing a k_{th} row in the undirected neighborhood matrix **B**. $\hat{\mathbf{B}}$ can be estimated for each k_{th} row separately based on the score function (9). Meinshausen and Buhlmann (2006) showed that $\hat{\mathbf{B}}$ through optimizing function (9) provides a consistent estimate of the conditional independence graph in high dimension. Equation (9) can be solved by the efficient pathwise coordinate optimization algorithm detailed in Friedman et al. (2007). We use *glmnet()* in R package.

It is known that if the generative distribution is faithful with respect to a DAG, the skeleton of the DAG is a subset of the corresponding conditional independence graph. We define $\hat{\mathbf{Ne}}$ as $\hat{\mathbf{Ne}}_{ij} = 1$ if $\hat{B}_{ij} \neq 0$, and $\hat{\mathbf{Ne}}$ estimated from Stage 1 is important to set up an parameter search space for stage 2. For sparse high dimensional graphs, Stage 1 greatly reduces the overall computational time, as the search space of Stage 2 is restricted within $\hat{\mathbf{Ne}}$.

3.2 Stage 2: DAG Estimation within the estimated Neighborhood

3.2.1 Single-Edge Update and DELTA Matrix—Before detailing the DAG estimation algorithm, we must first discuss minimizing the score function (6) given the parent matrix **Pa** = [Pa_{kj}]. $\mathbf{Pa}_k$ is the k_{th} row of **Pa**, and it denotes the set of parents of node X_k. For all k and j, $a_{kj} = 0$ if $Pa_{kj} = 0$; otherwise a_{kj} can be any value and needs to be estimated.

Given **Pa**, the objective function in (6) is separable (Shojaie and Michailidis, 2010), and there-fore it suffices to solve the optimization problem over each row of matrix **A**. Thus, the objective function (6) given **Pa** can be represented by

$S(\mathbf{A}|\mathbf{Pa})=\sum_{k=1}^{p}S_k(\mathbf{a}_k|\mathbf{Pa})=\sum_{k=1}^{p}S_k(\mathbf{a}_k|\mathbf{Pa}_k)$, and

$$S_k(\mathbf{a}_k|\mathbf{Pa}_k)=\frac{1}{n}\|\chi_k-\chi\mathbf{a}_k\|_2^2+\lambda\sum_{\{j|\mathrm{Pa}_{kj}=1\}}w_{kj}|a_{kj}|, \quad (10)$$

on the condition that $a_{kj} = 0$ if $Pa_{kj} = 0$; otherwise a_{kj} can be any value for all k and j. The score function value $S_k(\mathbf{a}_k|\mathbf{Pa}_k)$ and parameter value a_{kj} can be accurately calculated by the quadratic programming approach detailed in Supplementary Materials (Section 1). We use *optim()* in R package.

For convenience, we define the minimized value of the objective function, min $S_k(\mathbf{a}_k|\mathbf{Pa}_k)$, in (10) at $\mathbf{Pa}_k$ as a function of $\mathbf{Pa}_k$, $V_k(\mathbf{Pa}_k) = \min_{\mathbf{a}_k|\mathbf{Pa}_k} S_k(\mathbf{a}_k|\mathbf{Pa}_k)$. Therefore, min $S(\mathbf{A}|\mathbf{Pa})$ is a function of **Pa**, defined as

$$V(\mathbf{Pa})=\min_{\mathbf{A}|\mathbf{Pa}}S(\mathbf{A}|\mathbf{Pa}),$$

and $V(\mathbf{Pa})$ can be estimated by $V(\mathbf{Pa})=\sum_{k=1}^{p} V_k(\mathbf{Pa}_k)$.

Define $\mathbf{I}_{ij}$ as the diagonal matrix with the value one only in (i, j) entry. Therefore, $\mathbf{Pa} + \mathbf{I}_{ij}$ indicates adding one edge $i \leftarrow j$ in the current structure; in other words, adding one potential parent X_j for node X_i. We denote E_{ij} as the increment (negative value) of the minimized score function value by adding edge $i \leftarrow j$ from matrix **Pa**, so

$$E_{ij}=V(\mathbf{Pa}+\mathbf{I}_{ij}) - V(\mathbf{Pa}) \quad (11)$$

for edge $i \leftarrow j$ with $Pa_{ij} = 0$. Because V is separable, E_{ij} can be simply estimated by $E_{ij} = V_i(\mathbf{Pa}_i + \mathbf{I}_{ij}) - V_i(\mathbf{Pa}_i)$. Similarly, we use

$$L_{ij}=V(\mathbf{Pa}) - V(\mathbf{Pa} - \mathbf{I}_{ij}) \quad (12)$$

for edge $i \leftarrow j$ with $Pa_{ij} = 1$ to denote the decrement (positive value) in the minimized score function value by removing edge $i \leftarrow j$ from matrix **Pa**. L_{ij} can then be estimated by $L_{ij} = V_i(\mathbf{Pa}_i) - (\mathbf{Pa}_i - \mathbf{I}_{ij})$.

Matrix $\mathbf{E} = [E_{ij}]$ and $\mathbf{L} = [L_{ij}]$ are defined as the DELTA matrices. Therefore, the DELTA matrix **E** is used to denote the increment of the optimized score function value by adding each edge discretely from the current matrix **Pa**, while the DELTA matrix **L** is to denote the decrement of the optimized score function value by removing each edge discretely from the current matrix **Pa**.

In summary, we have presented in this subsection the essential element of the proposed algorithm. Given each **Pa** matrix, one can calculate the optimized objective function value $\min S(\mathbf{A}|\mathbf{Pa})$ with respect to matrix **A**, and thus estimate the increment or decrement of the optimized score function value by adding or removing one edge from the **Pa** matrix, denoted by the DELTA matrix. $S(\mathbf{A}|\mathbf{Pa})$ can be efficiently optimized over each row of matrix **A**.

3.2.2 Discrete Improving Search Algorithm with Tabu List (DIST Algorithm)—

The proposed DIST algorithm has two components: Discrete Improving Search (DIS) and Tabu list. The DIS algorithm is a heuristic algorithm with the search path following the steepest descent calculated based on the DELTA matrix. We take advantage of the properties that the skeleton of a DAG is the subset of the probabilistic neighborhood matrix when designing the DIS algorithm. In this way, the search space of selecting entering edges is greatly reduced. We start from an empty DAG, and the search space is upper bounded by $\hat{\mathbf{N}}\mathbf{e}$ from Stage 1. The algorithm is illustrated as follows.

1. Start: we start from an empty DAG, therefore $\mathbf{Pa}^0 = 0$.
2. Starting from t = 1, at each iteration we perform the following steps, called discrete improving search, to update $\mathbf{Pa}^t$ from $\mathbf{Pa}^{t-1}$:

a. Select an entering edge: calculate $E_{ij}^{t}=V(\mathbf{Pa}^{t-1}+\mathbf{I}_{ij})-V(\mathbf{Pa}^{t-1})$ for (i, j) satisfying the two criteria: (1) it is in the neighborhood matrix with $\hat{Ne}_{ij}\neq 0$, and (2) it was not selected in any previous iterations $Pa_{ij}^{t-1}=0$. We select an entering edge $X_i \leftarrow X_j$ with the most improvement of the objective function value E_{ij}^{t} among all the edges satisfying the two criteria. Then, we temporarily update $\mathbf{Pa}^{tmp}=\mathbf{Pa}^{t-1}+\mathbf{I}_{ij}$, and accordingly $\hat{\mathbf{A}}^{tmp}$ and $\hat{\mathbf{T}}^{\hat{\mathbf{A}}^{tmp}}$.

b. Check cycle(s): based on $\hat{\mathbf{T}}^{\hat{\mathbf{A}}^{tmp}}$, we check for the existence of any cycle and find the path of the cycle by checking Equation (1). This can be computationally simplified by forward and backward Breadth First Search (BFS) algorithms (Najork and Wiener, 2001; Kurant et al., 2010; West, 2001), detailed in Supplementary Materials (Section 2). If no cycle exists, we update $\mathbf{Pa}^{t}=\mathbf{Pa}^{tmp}$, $\hat{\mathbf{A}}^{t}=\hat{\mathbf{A}}^{tmp}$, and $\hat{\mathbf{T}}^{t}=\hat{\mathbf{T}}^{tmp}$ and skip Step (c). If at least one cycle exists, we proceed to Step (c).

c. Select leaving edge(s) if cycle(s) exists: for all edges $(\hat{\imath}, \hat{\jmath})$s on the cycle path, calculate $L_{\hat{\imath}\hat{\jmath}}^{t}=V(\mathbf{Pa}^{tmp})-V(\mathbf{Pa}^{tmp}-\mathbf{I}_{\hat{\imath}\hat{\jmath}})$, and select a leaving edge $X_{\tilde{\imath}} \leftarrow X_{\tilde{\jmath}}$ among all edges on the cycle path with the least decrement of the objective function value L_{ij}^{t}. To express this by mathematical notation, the leaving edge $X_{\tilde{\imath}} \leftarrow X_{\tilde{\jmath}}$ is selected with $E_{ij}^{t}<L_{\tilde{\imath}\tilde{j}^{t}<0}$ and $L_{\hat{\imath}\hat{\jmath}}^{t}\leq L_{\tilde{\imath}\tilde{j}^{t}}$ for any $(\hat{\imath}, \hat{\jmath})\in C$ where C is a set of edges in the cycle path. Note that more than one cycle might exist for any added single edge, so Steps (b) and (c) need to be applied repeatedly until no cycle exists. Let the set of leaving edges be $(\tilde{\imath}_h, \tilde{\jmath}_h)$. To confirm that the improvement of the objective function by adding edge $X_i \leftarrow X_j$ is higher than the sum of the decrements by all the leaving edges, we need to confirm if $E_{ij}^{t}<\sum_h L_{\tilde{\imath}_h\tilde{j}_h}{}^{t}$. If so, we confirm the entering edge (i, j) and all the selected leaving edges $(\tilde{\imath}_h, \tilde{\jmath}_h)$s, and then update $\mathbf{Pa}^{t}=\mathbf{Pa}^{t-1}+\mathbf{I}_{ij}-\sum_h \mathbf{I}_{\tilde{\imath}_h\tilde{\jmath}_h}$, and accordingly $\hat{\mathbf{A}}^{t}$ and $\hat{\mathbf{T}}^{t}$. If $E_{ij}^{t}\geq\sum_h L_{\tilde{\imath}_h\tilde{j}_h}{}^{t}$, the entering edge (i, j) can not be added; therefore, update $\mathbf{Pa}^{t}=\mathbf{Pa}^{t-1}$ and accordingly $\hat{\mathbf{A}}^{t}$ and $\hat{\mathbf{T}}^{t}$.

3. Repeat Step 2 until there is no edge for the improvement in the score function value, which means that no more edges can be entered or removed.

The DIS algorithm typically converges within a reasonable number of iterations. However, most local search algorithms, including the proposed DIS algorithm, check their neighboring solutions around a current solution to improve the score function value. These algorithms can be stuck in suboptimal solutions or saddle points, and the leaving edges often become reselected as an entering edge in the immediately succeeding iterations. To improve the efficiency of the algorithm, we incorporate it into the well-known metaheuristic search method, tabu search (Glover, 1989, 1990). A tabu search algorithm is designed to improve the performance of a local searching algorithm by using a memory queue (tabu list) where

the solutions previously selected within a short-term period are stored, then it prevents the algorithm from selecting the previous solution repeatedly (Glover, 1989, 1990). However, the original tabu search algorithm is applied after a greedy search algorithm finds a solution, as an additional algorithm. Thus, it is not clear how many additional iterations the tabu search algorithm needs to run, and how long the tabu list should be.

Here, we combine the DIS search algorithm with the tabu list, which is a modified version of the original tabu search technique, called the DIST algorithm. We build a global tabu list and store all leaving edges and all scanned edges in the list. After all edges are scanned and no edge remains to be searched, we temporarily stop running the algorithm. Then, we free the edges in

Algorithm 1

DIST algorithm

input : χ : the data matrix

λ : a penalty parameter

W : a weight matrix by $W = [w_{ij}]$

$\hat{\mathbf{Ne}}$: an estimated neighborhood matrix from stage 1

output : $\hat{\mathbf{A}}$: a coefficient matrix

initialization : Initialize $\mathbf{A}^0$ and $\mathbf{Pa}^0$ matrix by entering zeros in all entries.

Discrete Improving Search (DIS) Algorithm

repeat

Calculate $E_{ij}^t = V(\mathbf{Pa}^{t-1} + \mathbf{I}_{ij}) - V(\mathbf{Pa}^{t-1})$ for (i, j) with $Pa_{ij}^{t-1} = 0$ and $\hat{Ne}_{ij} = 1$

Select the entering edge (i, j) with $Pa_{ij}^{t-1} = 0$ in the condition that $E_{ij}^t < E_{\bar{i}\bar{j}}^t$ for any $(\bar{i}, \bar{j})$ with $Pa_{\bar{i}\bar{j}}^{t-1} = 0$ and $\hat{Ne}_{\bar{i}\bar{j}} = 1$;

Update $\mathbf{Pa}^{tmp} = \mathbf{Pa}^{t-1} + \mathbf{I}_{ij}$, and accordingly $\hat{\mathbf{A}}^{tmp}$, $\hat{\mathbf{T}}^{\hat{\mathbf{A}}^{tmp}}$;

if *cycle(s) exists in* $\hat{\mathbf{T}}^{\hat{\mathbf{A}}^{tmp}}$ *by* Equation (1) *or BFS* **then**

Calculate $L_{ij}^t = V(\mathbf{Pa}^{\mathrm{tmp}}) - V(\mathbf{Pa}^{\mathrm{tmp}} - \mathbf{I}_{ij})$ for (i, j) on the cycle path and $Pa_{ij}^{\mathrm{tmp}} = 1$;

Select a set of leaving edges $\{(\tilde{\imath}_h, \tilde{\jmath}_h) | h = 1,2, \ldots\}$, where $L_{\hat{\imath}_h\hat{\jmath}_h}^t \leq L_{\tilde{\imath}_h\tilde{\jmath}_h}^t$ for any $(\hat{\imath}_h, \hat{\jmath}_h) \in C_h$,

if $E_{ij}^t > \sum_h L_{\tilde{\imath}_h\tilde{\jmath}_h}^t$ **then**

Update $\mathbf{Pa}^t = \mathbf{Pa}^{t-1} + \mathbf{I}_{ij} - \sum_h \mathbf{I}_{\tilde{\imath}_h\tilde{\jmath}_h}$ and accordingly $\hat{\mathbf{A}}^t$ and $\hat{\mathbf{T}}^t$;

end

else $\mathbf{Pa}^t = \mathbf{Pa}^{t-1}$;

end

else Update $\mathbf{Pa}^t = \mathbf{Pa}^{t-1} + \mathbf{I}_{ij}$, and accordingly $\hat{\mathbf{A}}^t$;

Check $\hat{a}_{il} = 0$ for any parent l if $R_{il} = 1$. Then, update $\mathbf{Pa}^t$ by $Pa_{il} = 0$.

until $\forall(i, j), E_{i,j}^t > 0$ *with* $Pa_{ij}^t = 0$;

Tabu-augmented DIS: Build a dynamic tabu list in each round of Step 1 to avoid computational redundancy.

the tabu list, and repeat the search as the next round. If the score function value is not improved in the subsequent rounds, we stop and obtain the final solution. The succinct description of the DIST algorithm is shown in Algorithm 1.

After finishing the DIST algorithm, we obtain a DAG estimate. However, due to observational equivalence, there exist other DAG estimates that are in the same equivalence class. We can apply the algorithm discussed by Chickering (1995, 2002a), implemented in *essentialGraph*() function of R package *ggm*, to extend the estimated DAGs to a complete partial DAG (cpDAG). The algorithm essentially identifies the reversible edges in the DAG estimate to provide the same score function value, and it extends the DAGs to a complete partial DAG.

3.3 The Tuning Parameters

Different methods have been proposed for selecting the tuning parameter in the full conditional independence graphs or DAGs under known variable order. These include cross-validation procedures (Rothman et al., 2008), BIC-based selection (Yuan and Lin, 2007), and an α-based selection (Shojaie and Michailidis, 2010). Here we discuss the selection of λ_α based on Meinshausen and Buhlmann (2006), who propose $\lambda_\alpha = \frac{2\hat{\sigma}_{X_k}}{\sqrt{n}}\left[1 - \Phi^{-1}\left(\frac{\alpha}{2p^2}\right)\right]$ for controlling the error rate of the edge estimation and computational simplicity. Under known variable order, the asymptotic consistency is demonstrated (Shojaie and Michailidis, 2010) for $\lambda_k(\alpha) = \frac{2\hat{\sigma}_{X_k}}{\sqrt{n}}\left[1 - \Phi^{-1}\left(\frac{\alpha}{2p(k-1)}\right)\right]$, where $\Phi^{-1}()$ is the cumulative distribution function of $N(0, 1)$. Under the unknown ordering, we adapt the above λ_α or λ_k to reflect the possible candidates of $p - 1$ parents given a child k, and we also adjust the effect of the weights, which is

$$\lambda_n(\alpha) = \frac{2\hat{\sigma}_{X_k}}{\overline{w}\sqrt{n}}\left[1 - \Phi^{-1}\left(\frac{\alpha}{2p(p-1)}\right)\right], \quad (13)$$

where $\bar{w}$ is a constant related to the weight values w_{ij}. We define the constant $\bar{w}$ by the geometric average of the lower and upperbound of w_{ij}.

To show the asymptotic property, from (8), we rewrite the weight w_{ij} by

$$w_{ij} = \max(1, \min(\frac{1}{|\hat{\beta}_{ij}|^\gamma}, \frac{1}{(L_n)^\gamma})), \quad (14)$$

where $L_n = \min(1/N, 1/n)$. $(1/L_n)^\gamma$ can be interpreted as a threshold for the upper bound of w_{ij}, and as n goes to infinity, the threshold vanishes. Based on the formula of w_{ij} in 14, $1 < w_{ij} < \frac{1}{(L_n)^\gamma}$, so $\bar{w}$ is $\frac{1}{(L_n)^{(\gamma\cdot\delta)}}$, where $0 < \delta < 1$. Thus, the $\lambda_n(\alpha)$ in (13) can be rewritten by

$$\lambda_n(\alpha) = \frac{2L_n^{\gamma\cdot\delta}}{\sqrt{n}}\left[1 - \Phi^{-1}\left(\frac{\alpha}{2p(p-1)}\right)\right], \quad (15)$$

which satisfies the following condition.

Let $\tilde{\beta}_{ij} = \max(\hat{\beta}_{ij}, L_n)$ and β^* be the true value. Then, $\sqrt{n}(\tilde{\beta}_{ij} - \beta^*) = O_p(1)$, since $\hat{\beta}_{ij}$ is a lasso estimate, which is a root-n-consistent estimator to β^* (Zou, 2006), and $L_n = 1/n$ for $n > N$. In addition, $(n\lambda_n(\alpha))/\sqrt{n} = C_1(\frac{1}{n})^{\gamma\cdot\delta} \to 0$, and $(n\lambda_n(\alpha))n^{(r-1)/2} = C_2 n^{r(1-\delta)} \to \infty$, where C_1 and C_2 are constants. Thus, $\lambda_n(\alpha)$ gives a consistent variable selection based on Zou (2006). The consistent variable selection for each variable (each node) provides the probabilistic neighborhood asymptotically, which is a conditional independence graph (undirected graph) for sparse high dimensional graphs (Meinshausen and Buhlmann, 2006).

3.4 Advantages of the NS-DIST Method

The proposed NS-DIST method has several unique advantages.

- Statistical and computational efficiency

  The two-stage NS-DIST method provides an intuitive interpretation of the DAG estimation process, which first estimates an undirected graph or the neighborhoods, then estimates directionality within the estimated undirected graph. This is similar to a hybrid algorithm or Markov blanket. However, most hybrid methods combine two distinct search methods, which increase the number of parameters and model complexity. For example, the MMHC method uses a constraint-based approach to find the undirected graph, then applies a search-and-score approach to find the directionality. Our proposed method naturally implements the optimization in two steps but retains the same score criteria. In Stage 1, the method estimates the undirected graph by identifying a probabilistic neighborhood of each node. This approach only requires solving adaptive lasso regression for each node separately, which is a convex optimization problem. Thus, the neighborhood-search algorithm takes little computational time and can be implemented in parallel computing. In Stage 2, we utilize the estimated undirected graph as an upper bound of the search space of the directed graph, which greatly reduces the number of unknown parameters. This is especially important in high dimensional settings where brute-force DAG estimation is very challenging. In addition, the utility of the steepest descent search path and tabu list allows us to achieve convergence within a reasonable number of iterations.

Enabled by these computational efficiencies, the algorithm can estimate DAGs with 3000 parameters in about 2 CPU hours.

- Consistency in variable selection

  The neighborhood estimates from adaptive lasso in Stage 1 enjoy consistency for variable selection as established by Meinshausen and Buhlmann (2006) and Zou (2006), which means that the undirected graph estimated in Stage 1 converges to the true undirected graph. Furthermore, Chickering (2002b) demonstrated that if the generative distribution is faithful with respect to a DAG defined over the observable variables, a greedy search algorithm with an asymptotically consistent scoring criterion satisfies local consistency in DAG estimation in the limit of large sample sizes. The adaptive lasso is a consistent scoring criterion, so under conditions in Chickering (2002b), the NS-DIST method satisfies the local consistency.

- Robustness

  The proposed method uses an adaptive lasso regression as the score function. It has been discussed that adaptive lasso as a regularization approach gives more stable coefficient estimates in multicollinearity scenarios than those from OLS or ordinary lasso (Tibshirani, 1996; Zou, 2006; Hebiri and Lederer, 2013), as sparsity is imposed at different levels on each neighborhood. In contrast, for methods like PC-stable and MMHC, sparsity is utilized for the partial correlations as a whole view. As demonstrated in the simulation experiments, the adaptive lasso score function is more efficient and exible, especially for networks with hub structures.

- Flexibility

  A prior knowledge of network structures is often available or partially available. Thanks to the linear structural equation framework, prior knowledge can be incorporated quite exibly in our model. For example, we may assign some parameters of A_{ij} as known in terms of edges or directionality; or different weights w_{ij} according to their importance. Second, this method allows one to estimate the DAG structure around any variable of interest, so we do not have to estimate the global DAG structures for all the variables. This flexibility provides convenience in very high dimensional scenarios. Such an application will be demonstrated in Section 5.

4 SIMULATION STUDY

To assess the performance of the proposed method, we performed a series of simulation studies. We considered various simulation scenarios in terms of network topological structure, dimension-to-sample size ratio (p/n), signal-to-noise ratio, and normality. Edges are created according to either a random topological structure or a hub topological structure (Margolin et al., 2006). In a random network, edges are randomly created so that each node is equally likely to be connected to any other node. In a hub network, edges are created from a small number of hub nodes to their child nodes. The hub network structure mimics many

real biological networks in that only a small number of nodes are regulators and each has many targets to regulate (Margolin et al., 2006; Newman, 2003). To create a hub network, we first pick a certain proportion P_h of nodes to be hub nodes (parents' nodes), then simulate edges from the selected hub node to multiple child nodes. Unless Specified particularly, we set P_h = 10%. We use the sample size n=500 for all scenarios. We consider p = 50, 100, or 200 as a small p/n ratio, and also consider high dimensions such as p =1000, or even p=2000 or 3000 as large p/n ratios. The average density parameter d refers to the average number of edges per node, such that the total number of edges in a simulated network is $N = dp$. We use d=1 (indicating sparse network) or 2 (indicating dense network), and control the number of parents for any child to be between d−1 and d+1. For both random-structure and hub-structure networks, we consider a_{ij}=0.5 or 0.7 in **A** for low or high signal-to-noise ratio, respectively. The latent variables Z_ks are independently generated from a normal distribution with unequal standard deviations independently drawn from *Uniform*(0, 1). We also consider non-normal asymmetric distribution such as gamma distribution to investigate the performances of the methods under the violation of normality. The n observations corresponding to X_1, X_2, X_3, …, X_p are then generated according to the linear structural equation (2).

We compared the NS-DIST method with the following five recent DAG estimation methods: the PC-stable method provided by Colombo and Maathuis (2013), the MMHC method provided by Tsamardinos et al. (2006), the GES method provided by Chickering (2002b), the CD algorithm provided by Fu and Zhou (2013), and a permutation approach with the Lasso framework provided by Shojaie and Michailidis (2010). As proposed by Fu and Zhou (2013), we use γ = 0.15 for the CD method. We use λ_0 = 0.1 and γ = 0.15, the parameter controlling the weight in adaptive Lasso score functions for the NS-DIST, and permutation approach in formula (8). Discussion of how to obtain λ_0 and γ for the NS-DIST method is in Section 4 of the Supplementary Materials. We employed R packages *pcalg* for the PC-stable method and the GES method, *glmnet* for the permutation approach, and *bnlearn* for the MMHC method. We simulated 20 data sets under each combination of the parameters.

To examine the true and false positive rates (TPR and FPR), we adopt receiver operating characteristic (ROC) curves. TPR is calculated by $\frac{TP}{TE}$, where TP is the number of true positives or the number of true edges detected and TE is the number of true edges. FPR is calculated by $\frac{FP}{(p(p-1)/2 - TE)}$, where FP is the number of false positives or the number of edges claimed by mistake. ROC curves were estimated across a range of λ values for the NS-DIST method, the CD method, the GES method, and the permutation approach. For the PC-stable and MMHC methods, ROC curves were estimated over a range of α values. In addition, for directionality inference, we reported directional TPR (dTPR), defined as the number of detected true edges with their directions correctly estimated. As an illustration, if a graph contains a true directed edge $X \rightarrow Y$, we count the detection of such a directed edge as a directional true positive (dTP). In the case where the connection between X and Y is detected with the reverse direction ($X \leftarrow Y$), we count such an estimate as a directional false negative (dFN). If an estimated DAG contains only the connection without a Specified direction $X - Y$, we evaluate it as 1/2 dTP and 1/2 dFN. In addition, similar to the definition

of the false discovery rate (FDR), directional FDR (dFDR) is defined as $1-dTP/(TP+FP)$. We reported ROC curves with dTPRs plotted against FPRs for each method to examine directionality inference.

4.1 Comparison with other methods

To investigate the performance of the NS-DIST method, we compare it with other methods under low dimensional (p<n) and high-dimensional (p>n) settings. The ROC curves for p = 200 and n = 500 are shown in Figure 1, and those for p=50 and 100 show similar patterns (Section 6 of Supplementary Materials). The additional comparison plot based on network structure is in Section 7 of Supplementary Materials. The NS-DIST method (black solid lines) shows satisfactory performance in most cases and robustness with respect to network structures and densities. More Specifically, the NS-DIST method performs slightly better than the PC-stable (gray solid lines) and MMHC (gray long-dashed lines) methods under random structures in the range of small FPRs, but under the random and sparse (d=1) structure, the NS-DIST method is slightly out-performed by the two methods when FPR increases. The performance of the NS-DIST method remains robust under random and hub structures, while the performances of the PC-stable and MMHC methods decrease as more edges are concentrated from a smaller number of hub nodes (network range from random, 20% hub nodes to 10% hub nodes). Both methods use conditional independence tests based on partial correlations to identify edges. Perhaps problematically, the partial linear correlations tend to be unstable with presence of multicollinearity around the hub nodes, and the universal sparsity control parameter might generate overly sparse estimates around the hub nodes. On the other hand, the NS-DIST method may avoid those issues by employing an adaptive lasso score function. It has been discussed that adaptive lasso as a regularization approach gives more stable coefficient estimates in multicollinearity scenarios than those from OLS or ordinary lasso (Tibshirani, 1996; Zou, 2006; Hebiri and Lederer, 2013), as sparsity is imposed at different levels on each neighborhood.

Next, we compare the GES method (black dot-dashed lines) with the NS-DIST method. For the detection of edges, the NS-DIST method shows higher TPR or TPRs than the GES method given small FPRs, but as the FPR increases, the TPRs of the GES method become identical to those of the NS-DIST method. Under random and sparse network structures, the dTPRs of the GES become slightly better than those of the NS-DIST method as the FPR increases. The GES method uses the L_0 penalty, which leads to the hard-thresholding rule, whereas the adaptive lasso penalty yields the soft-thresholding rule. The variable selection solution using the L_0 penalty tends to be unstable compared to the continuous lasso shrinkage (Liu and Wu, 2007). Therefore in the ROC curves, the differences are most obvious with large penalty parameters, and reduce when the penalty parameters decrease.

Further, we compare the NS-DIST method with the CD method. Generally, the NS-DIST method shows higher TPRs than the CD method in small FPRs, which are obtained with larger penalty parameters. However, as the penalty parameter decreases, the gap between TPR (or dTPR) of the CD method and the NS-DIST method is reduced. Such performance patterns are observed in most scenarios. The CD algorithm is essentially a one-stage optimization algorithm using a penalized profile likelihood function with an adaptive Lasso

penalty term as its score function. This score function is still non-convex due to the log term; while our score function is a convex function without consideration of the acyclic restriction. The optimization of the non-convex score function is harder to implement, and a global optimization solution might be hard to achieve. Furthermore, the better performance of our method over the CD algorithm might come from the employment of the consistent neighborhood estimates from Stage 1.

We also compare the five methods under a smaller signal-to-noise ratio when a_{ij}=0.5, and the simulation results are in Section 8 of Supplementary Materials. All methods show less power than what we observe when a_{ij}=0.7. Although the comparison patterns among the methods are similar to Figure 1, the differences among the methods are smaller.

The ROC curves in high dimensional setting for p = 1000 and n = 500 are shown in Figure 2. The comparison patterns between the NS-DIST method and others when p=1000 are similar to those observed when p=200. The NS-DIST method shows satisfactory performance in most scenarios. However, the decrease in performance of the MMHC and PC-stable methods from random structure to hub structure is more severe in high dimensional settings. The gap between the NS-DIST algorithm and the GES algorithm in small ranges of FPRs also increased in dense networks compared to those observed when p = 200.

We then investigated the appropriate choice of λ for the NS-DIST method by examining FDR (dFDR) as well as MCC (dMCC) under a range of λ values as shown in Section 11 of Supplementary Materials. Noticeably, in most scenarios, the NS-DIST method produces reasonable performance with λ(α = 0.1, δ = 0.35). Therefore, we chose λ(α = 0.1, δ = 0.35) to examine the performance of the NS-DIST method in higher dimensions when p=2000 and 3000. In these scenarios, ROC curves are too time-consuming to be derived. To benchmark the performance of the NS-DIST algorithm, we compared it to the MMHC method - another two-stage based method. We select α=0.01 for the MMHC method to compare the performance of the NS-DIST method. The simulation results are noted in Table 1. The NS-DIST method shows significantly better performance in all metrics than the MMHC method. The dFDR are well controlled under 5% in all simulated scenarios. Even when p=3000, the computational time is about 2 CPU hours on average.

4.2 Computational Efficiency

Figure 3 shows computational time when p=400 or p=1000. The computational time is plotted over complexity of the estimated DAGs (presented as the number of estimated edges over p). As the plots show, the computational time of the NS-DIST method (black solid lines) is less than all other methods and is well-controlled when p or the estimated DAG complexity increases. This demonstrates that the two-stage strategy effectively reduces computational time. The computational time of the MMHC method (grey long-dashed lines) is close to that of the NS-DIST method, but grows significantly to estimate more complex DAGs. The computational time of the GES method is reasonable for all DAG complexities for lower-dimensional settings when p=400, but when p=1000, it increases exponentially as the complexity increases. A detailed analysis of computational complexity of the NS-DIST method is described in Section 10 of Supplementary Materials.

4.3 Additional Simulation Scenarios

To further investigate the performance of the NS-DIST method, we compared it with the adapted framework proposed by Shojaie and Michailidis (2010). Shojaie and Michailidis (2010) estimated DAGs assuming a known ordering input with the same score function as the proposed NS-DIST algorithm. For comparison purposes, we used two types of ordering inputs, true order and randomly permuted ordering. The estimate under the true ordering can be viewed as a theoretical upper bound of the NS-DIST method, while that under permutation ordering can be seen as a lower bound of the NS-DIST method. The simulation results are in Section 5 of Supplementary Materials. The dTPRs of the the NS-DIST method are within 80% of those from the true ordering approach for most cases, and significantly better than the permutation approach. Although around 10% ~ 40% of correct edge estimates were mis-detected with wrong directions for random and dense networks, the performance of the NS-DIST method still proved better than that of the permutation method.

We also compared the performance of the methods when the latent variables have non-normal skewed distribution, such as gamma distribution with various combinations of shape and scale parameters. The ROC curves of the methods are also shown in Section 9 of Supplementary Materials. The performances of all methods decrease under gamma distribution compared to normal distributed latent variables. However, the comparison patterns are similar to those under the normal distribution.

The NS-DIST method can be naturally adapted to accommodate the partially known ordering between nodes by restricting the coefficient matrix **A** with the known restrictions of directions or parameter values. Several algorithms have been proposed for the inference of causal relationships among phenotypes using genetic data and genomic data (Neto et al., 2008, 2010; Hageman et al., 2011). Here, we designed a simulation experiment to examine the impact of partially known ordering on the performance of DAG estimates. For this purpose, we simulated variables $U_1, \ldots, U_p$ independently from the standard normal distributions. X_i was simulated as before and was associated with one additional parent U_i. The entry in **A**, $a_{X_iU_i}$, is generated from a uniform distribution $U(0, 1)$. Such a simulation setup attempts to mimic the scenario where the DNA copy number directly affects the mRNA gene expressions, with $U_1, \ldots, U_p$ corresponding to copy numbers of genes 1, …, p, and $X_1, \ldots, X_p$ denoting expression of these genes. Note that while X_i can be driven by U_i (to the degree indicated by $a_{X_iU_i}$), it can also be regulated by other X_js. We applied the NS-DIST method to analyze the simulated coherent data $\chi_{n\times 2p}$ with partially-known ordering. Specifically, we assumed that directions from U_i to X_i are known as a prior information, but the directions within X_js are unknown for $i = 1, \ldots, p$. We then calculated an ROC curve based on estimated networks among X_js in order to make a fair comparison with the ROC curves based on previous DAG estimates among X_js, without involving U_js and corresponding known ordering. The ROC curves in Section 12 of Supplementary Materials demonstrate that although the additional information between U_i and X_i does not help in improving TPRs, such information slightly enhances the detection of directionality as indicated by dTPRs.

To investigate the optimization quality of the DIST algorithm, we further compared the DAG estimates from the DIST algorithm with those from global optima obtained by using the branch-and-bound technique under small dimension (d=10). The branch-and-bound technique optimizes the same score function in Equation (6). The detail of the search technique for global optimal solutions is described, and the comparison between the DIST algorithm and the upper bound of the branch-and-bound technique is shown in Supplementary Materials (Section 3). The difference in the optimized score function values from the DIST algorithm are within 0.06%~0.12% of those from the branch-and-bound technique. However, the running time of the branch-and-bound technique is much longer than that of the DIST algorithm. For example, for simulations with $p = 10$, $n = 200$, and $d = 2$, the average running time of the branch-and-bound technique implemented by CPLEX and C++ is 918.65 CPU seconds, and the running time of the DIST algorithm implemented by R is 1.03 CPU seconds. Therefore, these results suggest that the DIST algorithm can efficiently optimize the objective function with reasonable accuracy.

5 APPLICATIONS AND CASE STUDIES

5.1 Case study 1: Transcription Factor (TF) Networks in Ovarian Adenocarcinomas Tumor Samples

The Cancer Genome Atlas (TCGA) project has analyzed a wide range of genomic features in 429 high-grade serous ovarian adenocarcinomas (OV). However, since OV is a complex disease, there is a pressing need to study the networks between genes to better understand tumor progression, rather than analyzing individual genes. The Cancer Genome Atlas Research Network (2011) searched for altered pathways in the US National Cancer Institute Pathway Interaction Database (*PID*, http://pid.nci.nih.gov/, (Schaefer et al., 2009)), and found that the *FOXM1* transcription factor and its proliferation-related target genes, *AURKB*, *CCNB1*, *BIRC5*, *CDC25* and *PLK1*, are significantly activated by transcriptional regulation in OV samples. To potentially identify more TF pathways with activated transcriptional regulations, we analyzed the mRNA expressions of the 15 TF-encoding genes annotated in *PID*, including *JUN*, *JUND*, *JUNB*, *MYB*, *FOXA1*, *FOXA2*, *FOXA3*, *FOXM1*, *EPAS1*, *E2F1*, *E2F2*, *E2F3*, *E2F4*, *E2F5*, and *E2F6*. To identify their potential regulated targets, we extracted 147 genes that directly interact with the 15 TFs annotated in *NetBox* (http://cbio.mskcc.org/tools/netbox/index.html), which includes information from *PID*, *the Human Protein Reference Database* (Keshava Prasad et al., 2009), *Reactome* (Joshi-Tope et al., 2005; Matthews et al., 2009), and the MSKCC Cancer Cell Map (http://www.mskcc.org/). The level 3 normalized mRNA expressions of the 162 genes were obtained from OV tumor samples in the TCGA data portal (https://tcga-data.nci.nih.gov/tcga/dataAccessMatrix.htm). For each log-transformed mRNA expression, we first performed standardization by centralizing the mean to 0 and standard deviation to 1. We then applied the proposed NS-DIST algorithm and the existing algorithms to estimate the DAG structures among the 162 genes, with the ultimate goal of identifying the TFs as hub nodes and correctly identifying their corresponding regulated targets. *NetBox* annotations are used as the benchmark to decide the target genes of the TFs and the underlying network structure.

We first compared the methods based on their performance around the 15 hub TF nodes. For this purpose, we define the TPs of the TFs (TP_{TF}) as correctly estimated edges between TFs and their reported target genes, and the FPs of the TFs (FP_{TF}) as estimated edges between TFs and genes that were not annotated as their target genes in NetBox. Figure 4 (a) shows the ROC curves under a range of penalty parameters for each method. Importantly, one observes that the NS-DIST algorithm can detect more numbers of true regulatory connections than all other methods. We also compared all methods by investigating the TPs and FPs of the overall network structure of the 162 genes (Figure 4 (b)). In this case, the NS-DIST and CD estimates are very similar, and both are better than other algorithms.

In this example, we use the mRNA expressions of the 15 TF-encoding genes and the TF-target-protein-encoding genes to represent the protein activities of the TFs and their regulatory targets. One important assumption underlying this analysis is that the gene expressions are reasonable approximations of their corresponding protein expressions. To check this assumption, we correlated protein expression with encoding gene expression for the 106 protein expressions available in TCGA OV samples. Among those, 98.1%, 81.1% and 53.8% of the matched pairs showed Spearman correlations greater than 0, 0.2, and 0.5 respectively. With this data, we believe that the mRNA expressions are reasonable approximations of their corresponding protein expressions. Therefore, the NS-DIST algorithm has great potential to identify TFs with activated transcriptional regulation of their regulatory targets using mRNA data.

5.2 Case study 2: Drug Response Networks in Tumor Cell Lines

One unique advantage of the proposed NS-DIST method is its easy adaptation and modification for the purpose of identifying the sub-network around the variable of interest, with-out exhaustive estimation of the DAG of all variables. To illustrate this property, we applied the NS-DIST method to the Cancer Cell Line Encyclopedia (CCLE) data downloaded from http://www.broadinstitute.org, which contains a genomewide mRNA profile of 18,989 genes from a wide spectrum of human cancer cell lines (sample size of 491). These data also include their responses to several cancer treatment compounds, including the *MDM2* antagonist Nutlin (Barretina et al., 2012). Nutlin enables p53 to mediate its downstream functions, including activation of gene transcription and induction of cell cycle arrest and apoptosis (Vassilev et al., 2004). The ability to predict tumor responsiveness to the *MDM2* antagonist through mRNA biomarkers is an unmet need that could significantly improve clinical development. Therefore, the goal is to identify mRNAs predictive of the cell line responses to the *MDM2* antagonist and to detect the regulatory relationships among the identified mRNA predictors.

Since the entire dimension (18,989) is very high, we cannot accurately estimate the entire graph within reasonable computational time. Thus, by using the NS-DIST method, we tried to estimate the sub-network around the target drug response. First, we found the neighbors of *MDM2*, and recursively found neighbors of the previous ones until the total number of included genes was less than 2000. For selecting the penalty parameter λ, we use the formula (13) with $\alpha = 0.1$ and $\delta = 0.35$; to estimate weights, we use $\lambda_0 = 0.1$ and $\gamma = 0.15$, which are used in the simulation studies. Note that the direction between baseline

expressions and cell line responses to the MDM2 antagonist was presumed in the direction from expressions to cell line responses.

Here, we applied the first step of the NS-DIST method to find the predictive mRNAs of the MDM2-antagonist response (Y), then recursively applied it to find the neighbors of the newly identified mRNAs. Subsequently, given the genes selected from the above procedure, we applied the second step of the NS-DIST method to estimate network structure. Specifically, we estimated the DAG among the selected genes within the neighborhood structure, and we added the target response and its relationship to genes back to the estimated DAG. Figure 5 (a) shows the genes up to 3 edges away from the *MDM2*-antagonist response. In the plot, we colored the nodes depending on their functional processes: apoptosis (red), cell cycle arrest (yellow), cell growth (orange), DNA repair (blue), *MDM2*-related (cyan), *MDM2* (green), and the drug response Y (black). As shown in Figure 5 (a), many mRNAs impact *MDM2* antagonist response through various pathways. In addition, many genes in the sub-network were on the causal pathways to Y, because some of them were children of the same parent node (siblings of Y). We therefore refined the network into an ancestry sub-network containing only mRNAs with direct and indirect edges towards Y, and only show the two level neighbors around the drug response and *MDM2* (Figure 5 (b)). Seven out of 26 genes in the ancestry sub-network are in biologically important pathways of the *MDM2*-antagonist, which retained the enrichment of biologically relevant genes from the *MDM2* sub-network. In order to quantify the statistical significance of the pathway enrichment levels in the sub-network, we calculated the p-value from Fisher's exact test based on the contingency table. The number of *MDM2*-related important genes is 53 out of 18,989 genes. The p-value for the sub-network in Figure 5 (b) is 5.5×10^{-13}, which is a significantly small value. This example shows that the NS-DIST method is able to effectively estimate a sub-network in ultra high dimensional data.

6 DISCUSSION

In this paper, we have discussed estimation of DAGs using a penalized likelihood approach with an adaptive Lasso penalty without knowing the ordering of the variables. The acyclicity constraint on the structure of DAGs poses a challenge to the optimization of the score function. We propose a two-stage adaptive lasso approach, the NS-DIST method, which first estimates an undirected graph by the neighborhood selection approach (Stage 1), and then estimates the DAG within the selected neighborhood (Stage 2) via the DIST algorithm, which is a meta-heuristic solution search method. The two stages use the same score function, which is a penalized likelihood function with an adaptive Lasso penalty term. The NS-DIST method is statistically and computationally efficient and has shown satisfactory performance in simulation experiments and real data examples.

Compared to the two-stage approach, one could alternatively apply the DIST algorithm brute force on the whole parameter space without the neighbored selection step. This single-stage algorithm is very computationally consuming. As demonstrated in Section 13 of the Supplementary Materials, we find that the single-stage algorithm achieved almost identical or slightly higher TPRs than those from the two-stage algorithm, but with higher FPRs in most scenarios. One reason might be that the unrestricted parameter space of the DIST

algorithm in the single-stage approach is more prone to FPs, while for the two stage algorithm, the search space of the NS-DIST method is restricted in the estimated neighborhood with asymptotic consistency (Meinshausen and Buhlmann, 2006).

In high dimensional variable selection, other modified approaches, such as the smoothly clipped absolute deviation penalty (SCAD, Fan and Li (2001)) and the minimax concave penalty (MCP, Zhang (2010)), have been proposed. However, because the SCAD and MCP are nonconvex in the domain of the coefficient a_{ji}, it is not feasible to find the global optimum solution in terms of the score function, even without acyclic constraints. Thus, when we use such penalties under our two-stage approach, Stage 1 can provide only one local optimal solution. Kim et al. (2008) proved that one local optimal solution from SCAD satisfies the consistency in variable selection under relaxed conditions, but it is still questionable how to identify the consistent estimator among local optimal solutions (Wang et al., 2013). Additionally, Zhang (2010) argued that MCP can find the consistent estimator with its developed PLUS algorithm under certain regularity conditions. Still, further study is needed to investigate the consistent probabilistic neighborhood and estimation of DAGs with such penalty functions.

Chickering (2002a) discussed that searching among equivalence classes of network structures (E-space) is more efficient than searching among individual DAG structures (B-space). They demonstrated that a concern with using B-space is computationally inefficiency, as searching traverses within an equivalence class (Chickering, 2002a). The L_1 penalized likelihood score functions are score equivalent. The proposed DIST algorithm is a metaheuristic algorithm that searches among individual DAG structures. However, the loss of efficiency is not severe because each step of the DIST algorithm follows the steepest descent of the score function value. There-fore, it ensures an improvement of the score function at each search step, without wasting time traversing among DAG structures within an equivalence class. It would be interesting to adapt the set of operators, introduced by Chickering (2002a), that can be applied to move among equivalence classes in the framework of the DIST algorithm. This has the potential to further optimize computational efficiency.

Although we have focused on Gaussian variables in this study, our approach may be extended to other distributions, especially for count data generated by RNAseq platforms. It would be interesting to adapt the L_1 penalty framework in generalized linear regressions to estimate the structures of DAGs among nonnormal distributed data, though the optimization would be more challenging, especially in the high dimensional context.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

We also thank Dr. Colombo and Dr. Maathuis for providing the PC-stable algorithm codes. Research is supported by NIH-1-R21 GM110450-01.

References

Aragam B, Zhou Q. Concave Penalized Estimation of Sparse Gaussian Bayesian Networks. Journal of Machine Learning Research. 2015 in press.

Barretina J, Caponigro G, Stransky N, Venkatesan K, Margolin AA, Kim S, Wilson CJ, Lehar J, Kryukov GV, Sonkin D, Reddy A, Liu M, Murray L, Berger MF, Monahan JE, Morais P, Meltzer J, Korejwa A, Jane-Valbuena J, Mapa FA, Thibault J, Bric-Furlong E, Raman P, Shipway A, Engels IH, Cheng J, Yu GK, Yu J, Aspesi P, de Silva M, Jagtap K, Jones MD, Wang L, Hatton C, Palescandolo E, Gupta S, Mahan S, Sougnez C, Onofrio RC, Liefeld T, MacConaill L, Winckler W, Reich M, Li N, Mesirov JP, Gabriel SB, Getz G, Ardlie K, Chan V, Myer VE, Weber BL, Porter J, Warmuth M, Finan P, Harris JL, Meyerson M, Golub TR, Morrissey MP, Sellers WR, Schlegel R, Garraway LA. The Cancer Cell Line Encyclopedia Enables Predictive Modelling of Anticancer Drug Sensitivity. Nature. 2012; 483:603–607. [PubMed: 22460905]

Beal MJ, Falciani F, Ghahramani Z, Rangel C, Wild DL. A Bayesian Approach to Reconstructing Genetic Regulatory Networks with Hidden Factors. Bioinformatics. 2005; 21:349–356. [PubMed: 15353451]

Breiman L. Better Subset Regression Using the Nonnegative Garotte. Technometrics. 1995; 37:373–384.

Broom, BM., Subramanian, D., Vannucci, M. Computational Methods for Learning Bayesian Networks From High-Throughput Biological Data in Bayesian Inference for Gene Expression and Proteomics. Do, K-A., Muller, P., editors. New York: Cambridge University Press; 2006.

Buntine WL. Operations for Learning With Graphical Models. Journal of Artificial Intelligence Research. 1994; 2:159–225.

Buntine WL. A Guide to the Literature on Learning Probabilistic Networks From Data. IEEE Transactions on Knowledge and Data Engineering. 1996; 8:195–210.

Butte AJ, Tamayo P, Slonim D, Golub TR, Kohane IS. Discovering Functional Relationships Between RNA Expression and Chemotherapeutic Susceptibility Using Relevance Networks. Proceedings of the National Academy of Sciences of the United States of America. 2000; 97:12182–12186. [PubMed: 11027309]

The Cancer Genome Atlas Research Network. Integrated Genomic Analyses of Ovarian Carcinoma. Nature. 2011; 474:609–615. [PubMed: 21720365]

Chickering, DM. A Transformational Characterization of Equivalent Bayesian Network Structures. Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence, Montreal, QU, 87–98; Morgan Kaufmann. 1995.

Chickering DM. Learning Equivalence Classes of Bayesian-Network Structures. Journal of Machine Learning Research. 2002a; 2:445–498.

Chickering DM. Optimal Structure Identification With Greedy Search. Journal of Machine Learning Research. 2002b; 3:507–554.

Colombo, D., Maathuis, MH. Order-Independent Constraint-Based Causal Structure Learning. Technical Report. 2013. (arXiv:1211.3295v2)

Daly R, Shen Q, Aitken S. Learning Bayesian Networks: Approaches and Issues. The Knowledge Engineering Review. 2011; 26:99–157.

Fan J, Li R. Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties. Journal of the American Statistical Association. 2001; 96:1348–1360.

Ferrara CT, Wang P, Neto EC, Stevens RD, Bain JR, Wenner BR, Ilkayeva OR, Keller MP, Blasiole DA, Kendziorski C, Yandell BS, Newgard CB, Attie AD. Genetic Networks of Liver Metabolism Revealed by Integration of Metabolic and Transcriptomic Profiling. PLoS Genetics. 2008; 4:e1000034. PMCID: PMC2265422. [PubMed: 18369453]

Friedman N, Linial M, Nachman I, Pe'er D. Using Bayesian Networks to Analyze Expression Data. Journal of Computational Biology. 2000; 7:601–620. [PubMed: 11108481]

Friedman J, Hastie T, Hoeing H, Tibshirani R. Pathwise Coordinate Optimization. The Annals of Applied Statistics. 2007; 2:302–332.

Friedman J, Hastie T, Tibshirani R. Sparse Inverse Covariance Estimation With the Graphical Lasso. Biostatistics. 2008; 9:432–441. [PubMed: 18079126]

Fu F, Zhou Q. Learning Sparse Causal Gaussian Networks With Experimental Intervention: Regularization and Coordinate Descent. Journal of the American Statistical Association. 2013; 108:288–300.

Glover F. Tabu Search-Part I. ORSA Journal on Computing. 1989; 1:190–206.

Glover F. Tabu Search-Part II. ORSA Journal on Computing. 1990; 2:4–32.

Guyon I, Elisseeff A. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research. 2003; 3:1157–1182.

Ha MJ, Sun W, Xie J. PenPC: A two-step approach to estimate the skeletons of high-dimensional directed acyclic graphs. Biometrics. 2015

Hageman RS, Leduc MS, Korstanje R, Paigen B, Churchill GA. A Bayesian Framework for Inference of the Genotype-Phenotype Map for Segregating Populations. Genetics. 2011; 187:1163–1170. [PubMed: 21242536]

Hastie, T., Tibshirani, R., Friedman, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer; 2009.

Hauser A, Bühlmann P. Characterization and Greedy Learning of Interventional Markov Equivalence Classes of Directed Acyclic Graphs. Journal of Machine Learning Research. 2012; 13:2409–2464.

Hebiri M, Lederer JC. How Correlations Influence Lasso Prediction. IEEE Transactions on Information Theory. 2013; 59:1846–1854.

Heckerman, D. A Tutorial on Learning With Bayesian Networks. Technical Report MSR-TR-95-06, Microsoft Research. 1995.

Heckerman D, Geiger D, Chickering D. Learning Bayesian Networks: the Combination of Knowledge and Statistical Data. Machine Learning. 1995; 20:197–243.

Imoto S, Kim S, Goto T, Miyano S, Aburatani S, Tashiro K, Kuhara S. Bayesian Network and Nonparametric Heteroscedastic Regression for Nonlinear Modeling of Genetic Network. Journal of Bioinformatics Computational Biology. 2003; 1:231–252. [PubMed: 15290771]

Jansen R, Yu H, Greenbaum D, Kluger Y, Krogan NJ, Chung S, Emili A, Snyder M, Greenblatt JF, Gerstein M. A Bayesian Networks Approach for Predicting Protein-Protein Interactions From Genomic Data. Science. 2003; 302:449–453. [PubMed: 14564010]

John GH, Kohavi R, Pfleger K. Irrelevant Feature and the Subset Selection Problem. Proceedings of the Eleventh International Conference on Machine Learning. 1994:121–129.

Joshi-Tope G, Gillespie M, Vastrik I, D'Eustachio P, Schmidt E, de Bono B, Jassal B, Gopinath GR, Wu GR, Matthews L, Lewis S, Birney E, Stein L. Reactome: A knowledgebase of biological pathways. Nucleic acids research. 2005; 33:428–432.

Kalisch M, Bühlmann P. Estimating High-Dimensional Directed Acyclic Graphs With the PC-Algorithm. The Journal of Machine Learning Research. 2007; 8:613–636.

Kalisch M, Fellinghauer BAG, Grill E, Maathuis MH, Mansmann U, Bühlmann P, Stucki G. Understanding Human Functioning Using Graphical Models. BMC Medical Research Methodology. 2010; 10:14. [PubMed: 20149230]

Kalisch M, Machler M, Colombo D, Maathuis M, Bühlmann P. Causal Inference Using Graphical Models With the R Package pcalg. Journal of Statistical Software. 2012; 47:1–26.

Keller MP, Choi YJ, Wang P, Davis DB, Rabaglia ME, Oler AT, Stapleton DS, Argmann C, Schueler KL, Edwards S, Steinberg HA, Neto EC, Kleinhanz R, Turner S, Hellerstein MK, Schadt EE, Yandell BS, Kendziorski CM, Attie AD. A Gene Expression Network Model of Type 2 Diabetes Establishes a Relationship Between Cell Cycle Regulation in Islets and Diabetes Susceptibility. Genome Research. 2008; 18:706–716. [PubMed: 18347327]

Keshava Prasad TS, Goel R, Kandasamy K, Keerthikumar S, Kumar S, Mathivanan S, Telikicherla D, Raju R, Shafreen B, Venugopal A, Balakrishnan L, Marimuthu A, Banerjee S, Somanathan DS, Sebastian A, Rani S, Ray S, Harrys Kishore CJ, Kanth S, Ahmed M, Kashyap MK, Mohmood R, Ramachandra YL, Krishna V, Rahiman BA, Mohan S, Ranganathan P, Ramabadran S, Chaerkady R, Pandey A. Human protein reference database–2009 update. Nucleic acids research. 2009; 37(Database issue):767–772.

Kalisch M, Machler M, Colombo D, Maathuis M, Bühlmann P. Causal Inference Using Graphical Models With the R Package pcalg. Journal of Statistical Software. 2012; 47:1–26.

Kim Y, Choi H, Oh H-S. Smoothly Clipped Absolute Deviation on High Dimensions. Journal of the American Statistical Association. 2008; 103:1665–1673.

Liu Y, Wu Y. Variable Selection via a Combination of the L0 and L1 Penalties. Journal of Computational and Graphical Statistics. 2007; 16:782–798.

Kurant M, Markopoulou A, Thiran P. On the Bias of BFS (Breadth First Search). International Teletraffic Congress (ITC 22. 2010:1–8.

Lauritzen, SL. Graphical Models. Oxford: Clarendon Press; 1996.

Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera RD, Califano A. ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. BMC Bioinformatics. 2006; 7(Suppl 1):S7.

Markowetz F, Spang R. Inferring Cellular Networks - A Review. BMC Bioinformatics. 2007; 8(Suppl 6):S5.

Materna SC, Oliveri P. A Protocol for Unraveling Gene Regulatory Networks. Nature Protocols. 2008; 12:1876–1887.

Meinshausen N, Bühlmann P. High-Dimensional Graphs and Variable Selection With the Lasso. The Annals of Statistics. 2006; 34:1436–1462.

Matthews L, Gopinath G, Gillespie M, Caudy M, Croft D, de Bono B, Garapati P, Hemish J, Hermjakob H, Jassal B, Kanapin A, Lewis S, Mahajan S, May B, Schmidt E, Vastrik I, Wu G, Birney E, Stein L, D'Eustachio P. Reactome knowledgebase of human biological pathways and processes. Nucleic acids research. 2009; 37(Database issue):619–622.

Nagarajan R, Datta S, Scutari M, Beggs M, Nolen G, Peterson C. Functional Relationships Between Genes Associated With Differentiation Potential of Aged Myogenic Progenitors. Frontiers in Physiology. 2010; 1:1–8. [PubMed: 21522484]

Najork M, Wiener JL. Breadth-First Search Crawling Yields High-Quality Pages. Proceedings of the 10th International Conference on World Wide Web. 2001:114–118.

Neapolitan, RE. Series in Artificial Intelligence. Prentice Hall; 2004. Learning Bayesian Networks.

Neto EC, Ferrara CT, Attie AD, Yandell BS. Inferring Causal Phenotype Networks From Segregating Populations. Genetics. 2008; 179:1089–1100. [PubMed: 18505877]

Neto EC, Keller MP, Attie AD, Yandell BS. Causal Graphical Models in Systems Genetics: A Unified Framework for Joint Inference of Causal Network and Genetic Architecture for Correlated Phenotypes. The Annals of Applied Statistics. 2010; 4:320–339. [PubMed: 21218138]

Newman MEJ. The Structure and Function of Complex Networks. SIAM Review. 2003; 45:167–256.

Oldham M, Horvath S, Geschwind D. Conservation and Evolution of Gene Coexpression Networks in Human and Chimpanzee Brains. Proceedings of the National Academy of Sciences of the United States of America. 2006; 103:17973–17978. [PubMed: 17101986]

Pellet J-P, Elisseeff A. Using Markov Blankets for Causal Structure Learning. The Journal of Machine Learning Research. 2008; 9:1295–1342.

Pearl, J. Causality: Models, Reasoning, and Inference. Cambridge: Cambridge University Press; 2000.

Ramsey, J. Technical Report, CMUPHIL-177. Carnegie Mellon University; 2006. A PC-Style Markov Blanket Search for High Dimensional Datasets. Available at http://www.hss.cmu.edu/philosophy/techreports/177_Ramsey.pdf

Raskutti, G., Uhler, C. Learning Directed Acyclic Graphs Based on Sparsest Permutations. Technical Report. 2013. Available at arXiv:1307:0366

Robinson, R. Counting Unlabeled Acyclic Digraphs. In: Little, CHC., editor. Combinatorial Mathematics. V. Proceedings of Fifth Australian Conference Held at the Royal Melbourne Institute of Technology. Berlin: Springer; 1977. p. 28-43.

Ross, SM. Stochastic Processes. New York: John Wiley & Sons, Inc.; 1996.

Rothman AJ, Bickel PJ, Levina E, Zhu J. Sparse Permutation Invariant Covariance Estimation. Electronic Journal of Statistics. 2008; 2:494–515.

Schaefer CF, Anthony K, Krupa S, Buchoff J, Day M, Hannay T, Buetow KH. PID: The pathway interaction database. Nucleic acids research. 2009; 37(Database issue):674–679.

Schmidt, M., Niculescu-Mizil, A., Murphy, K. Proceeding of the 22nd AAAI Conference on Artificial Intelligence. Vol. 7. Menlo Park: AAAI Press; 2007. Learning Graphical Model Structure Using L1-Regularization Paths; p. 1278-1283.

Shao J. An Asymptotic Theory for Linear Model Selection. Statistica Sinica. 1997; 7:221–264.

Shi P, Tsai CL. Regression Model Selection: a Residual Likelihood Approach. Journal of the Royal Statistical Society. Series B (Statistical Methodology). 2002; 64:237–252.

Shojaie A, Michailidis G. Penalized Likelihood Methods for Estimation of Sparse High-Dimensional Directed Acyclic Graphs. Biometrika. 2010; 97:519–538. [PubMed: 22434937]

Spirtes, P., Glymour, C., Scheines, R. Causation, Prediction, and Search. Cambridge: MIT Press; 2000.

Stekhoven DJ, Moraes I, Sveinbjörnsson G, Hennig L, Maathuis MH, Bühlmann P. Causal Stability Ranking. Bioinformatics. 2012; 22:2819–2823.

Steuer R. On the Analysis and Interpretation of Correlations in Metabolomic Data. Briefings in Bioinformatics. 2006; 7:151–158. [PubMed: 16772265]

Tibshirani R. Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Statistical Methodology). 1996; 58:267–288.

Tsamardinos I, Brown L, Aliferis C. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. Machine Learning. 2006; 65:31–78.

Tu Z, Keller MP, Zhang C, Rabaglia ME, Greenawalt DM, Yang X, Wang IM, Dai H, Bruss MD, Lum PY, Zhou YP, Kemp DM, Kendziorski C, Yandell BS, Attie AD, Schadt EE, Zhu J. Integrative Analysis of a Cross-loci Regulation Network Identifies App as a Gene Regulating Insulin Secretion from Pancreatic Islets. PLoS Genetics. 2012; 8:e1003107. [PubMed: 23236292]

Vassilev LT, Vu BT, Graves B, Carvajal D, Podlaski F, Filipovic Z, Kong N, Kammlott U, Lukacs C, Klein C, Fotouhi N, Liu EA. In Vivo Activation of the p53 Pathway by Small-Molecule Antagonists of MDM2. Science. 2004; 303:844–848. [PubMed: 14704432]

Wang L, Kim Y, Li R. Calibrating Nonconvex Penalized Regression in Ultra-high Dimension. The Annals of Statistics. 2013; 41:2505–2536. [PubMed: 24948843]

Werhli AV, Grzegorczyk M, Husmeier D. Comparative Evaluation of Reverse Engineering Gene Regulatory Networks With Relevance Networks, Graphical Gaussian Models and Bayesian Networks. Bioinformatics. 2006; 22:2523–2531. [PubMed: 16844710]

West, DB. Introduction to Graph Theory. 2nd. Upper Saddle River: Prentice-Hall, Inc.; 2001.

Yuan, Y., Shen, X., Pan, W. Technical Report. University of Minnesota; 2012. Maximum Likelihood Estimation Over Directed Acyclic. Available at: http://www.sph.umn.edu/faculty1/wpcontent/uploads/2012/11/rr2012-013.pdf

Yuan M, Lin Y. Model Selection and Estimation in the Gaussian Graphical Model. Biometrika. 2007; 94:19–35.

Zhang CH. Nearly Unbiased Variable Selection under Minimax Concave Penalty. The Annals of Statistics. 2010; 38:894–942.

Zou H. The Adaptive Lasso and Its Oracle Properties. Journal of the American Statistical Association. 2006; 101:1418–1429.

Zou M, Conzen SD. A New Dynamic Bayesian Network (dbn) Approach for Identifying Gene Regulatory Networks From Time Course Microarray Data. Bioinformatics. 2005; 21:71–79. [PubMed: 15308537]

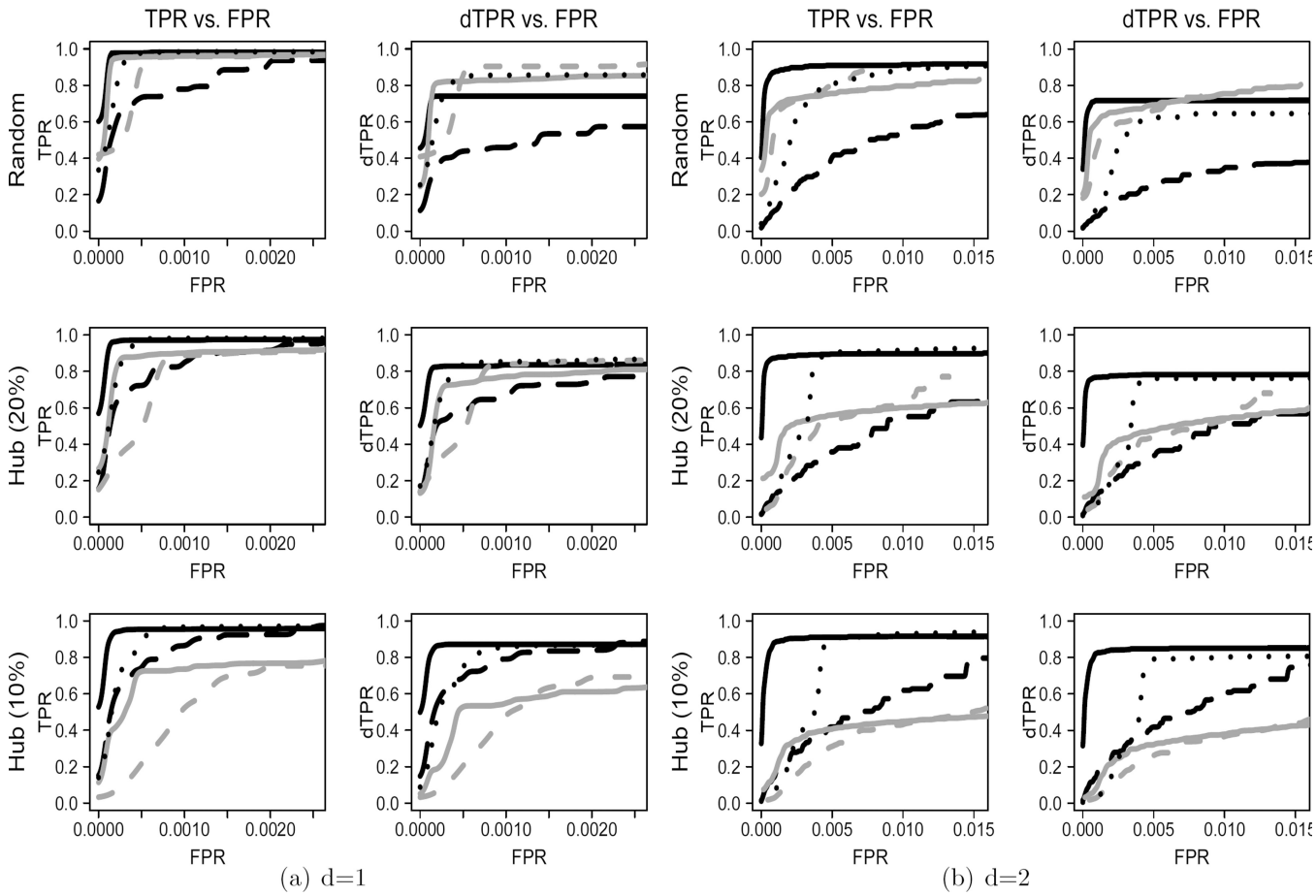


Figure 1.

ROC curves for $p = 200$ and $a_{ij} = 0.7$: The black solid line indicates the NS-DIST method, the black long-dashed line indicates the CD method, and the black dot-dashed line indicates the GES method. The gray solid line indicates the PC-stable method, and the gray long-dashed line indicates the MMHC method. The sample size n is 500

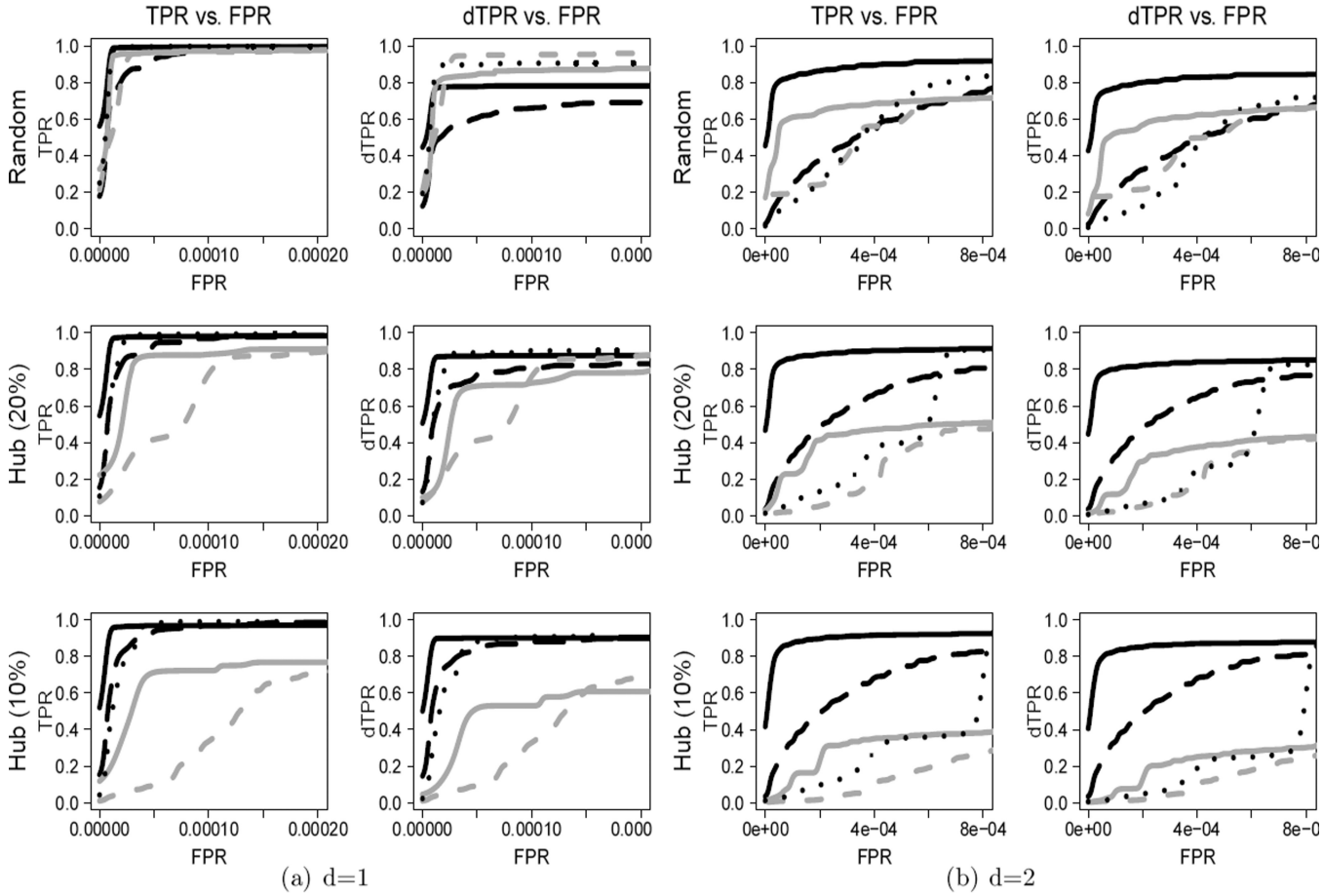


Figure 2.

ROC curves for $p = 1000$ and $a_{ij} = 0.7$: The black solid line indicates the NS-DIST method, the black long-dashed line indicates the CD method, and the black dot-dashed line indicates the GES method. The gray solid line indicates the PC-stable method, and the gray long-dashed line indicates the MMHC method. The sample size n is 500

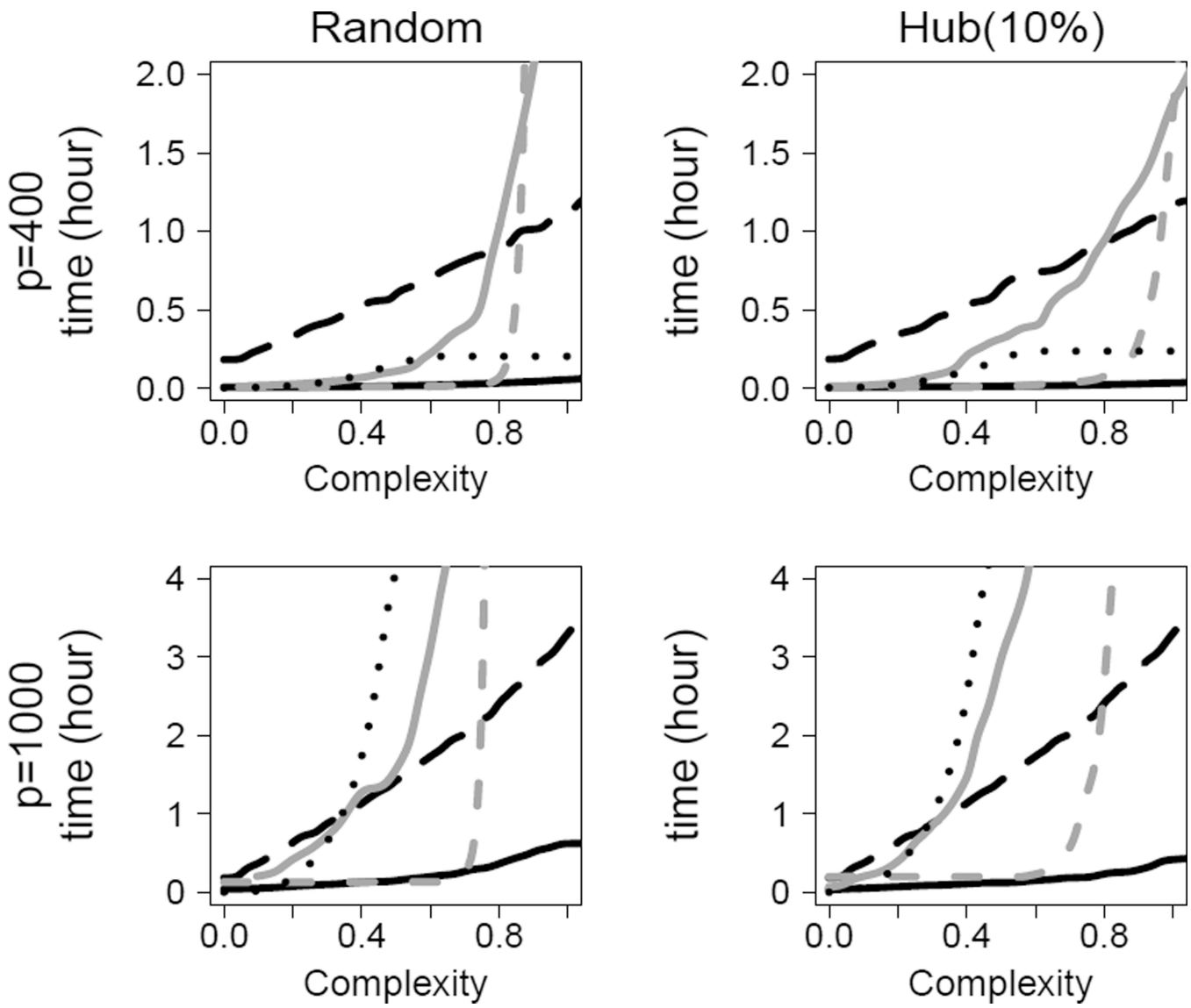
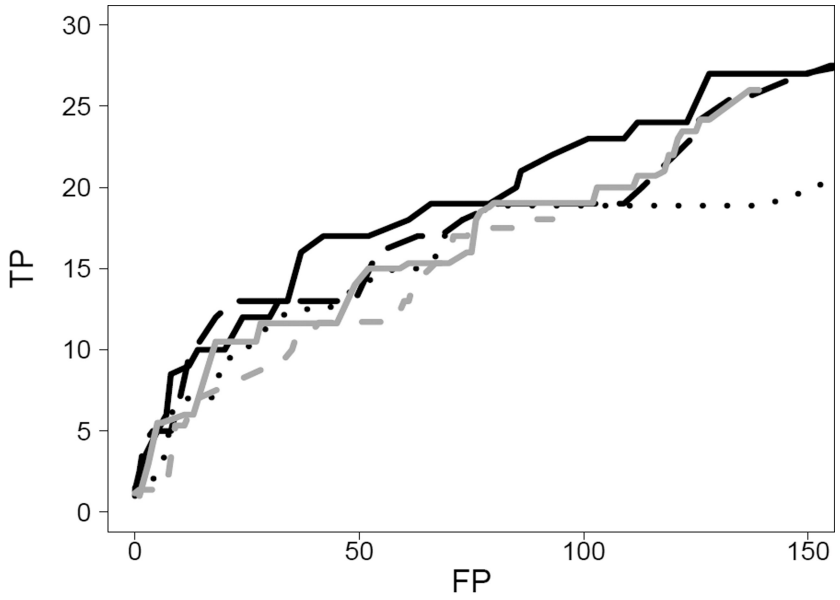
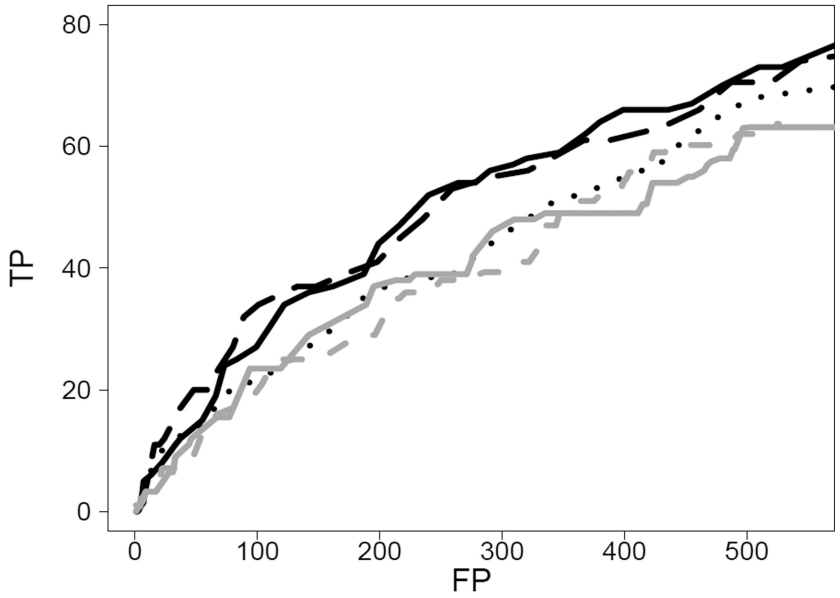


Figure 3.

Computational time vs. complexity (the number of estimated edges divided by p) for $d = 2$, random network, and $a_{ij} = 0.7$: The black solid line indicates the NS-DIST method, the dashed black line indicates the CD method, and the black dot-dashed line indicates the GES method. The solid gray line indicates the PC-stable method, and the dashed gray line indicates the MMHC method.



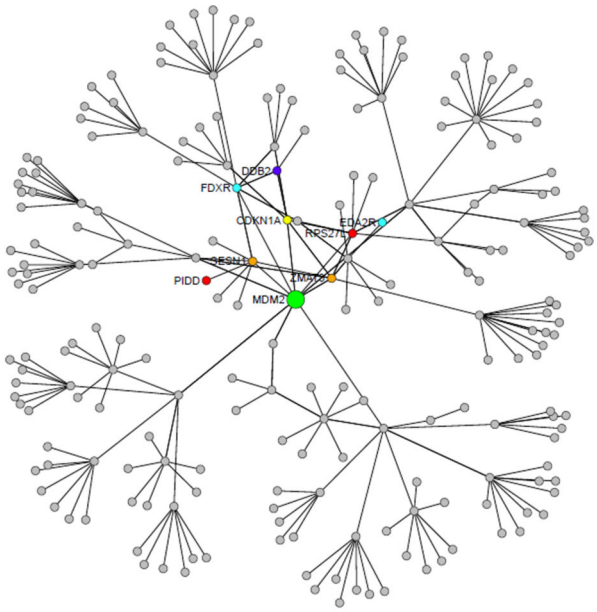
(a) ROC curves based on TP_{TF} and FP_{TF} around the hub nodes



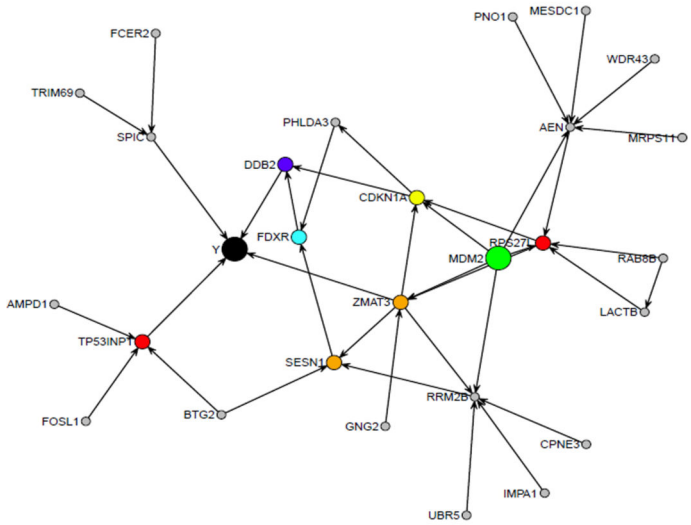
(b) ROC curves based on TP and FP of the overall network

Figure 4.

The result of network estimation from TCGA OV Tumors: Figure (a) shows ROC curves based on TP_{TF} and FP_{TF} around the hub nodes, and Figure (b) shows ROC curves based on TP and FP of the overall network. The black solid line indicates the NS-DIST method, the dashed black line indicates the CD method, and the black dot-dashed line indicates the GES method. The solid gray line indicates the PC-stable method, and the dashed gray line indicates the MMHC method.



(a) Undirected network around *MDM2* gene



(b) Gene network with ancestors of drug response Y

Figure 5.

CCLE data: The thickness of the edges indicates the absolute value of the estimated coefficient. Colors of the nodes depend on their functional processes, which are apoptosis (red), cell cycle arrest (yellow), cell growth (orange), DNA repair (blue), *MDM2*-related (cyan), *MDM2* (green), and *MDM2* response, Y (black).

Table 1

Summary of the simulation results of the NS-DIST method and the MMHC method in high dimension when d=2 and sample size n=500.

p	Type	Method	Parameter	TP	dTP	FP	FDR	dFDR	CPU time(sec.)
2000	Random	NS-DIST	0.30736	2559.2	2429.3	9.4	0.004	0.054	3429
	Random	MMHC	0.01	2397.2	2138.6	238.1	0.090	0.188	2870
	Hub(10%)	NS-DIST	0.30736	3163.5	3051.0	22.1	0.007	0.042	3605
	Hub(10%)	MMHC	0.01	1418.5	1280.7	1096.1	0.436	0.491	5551
3000	Random	NS-DIST	0.31506	3621.0	3470.9	48.7	0.015	0.056	7092
	Random	MMHC	0.01	3530.2	3156.7	350.6	0.090	0.187	12501
	Hub(10%)	NS-DIST	0.31506	4519.7	4391.6	38.4	0.009	0.037	8048
	Hub(10%)	MMHC	0.01	2089.7	1911.1	1575.6	0.430	0.479	24451