Received: 6 May 2023 | Revised: 25 March 2024 | Accepted: 18 May 2024

DOI: 10.1002/sim.10125

RESEARCH ARTICLE

Statistics in Medicine WILEY

# Detecting responsible nodes in differential Bayesian networks

Xianzheng Huang[1] | Hongmei Zhang[2]

[1]Department of Statistics, University of South Carolina, Columbia, South Carolina, USA

[2]Division of Epidemiology, Biostatistics, and Environmental Health, School of Public Health, University of Memphis, Memphis, Tennessee,

**Correspondence**
Xianzheng Huang, Department of Statistics, University of South Carolina, Columbia, SC 29208, USA.
Email: huang@stat.sc.edu

**Funding information**
National Institute of Allergy and Infectious Diseases, Grant/Award Number: R01AI121226

To study the roles that different nodes play in differentiating Bayesian networks under two states, such as control versus disease, we formulate two node-specific scores to facilitate such assessment. The first score is motivated by the prediction invariance property of a causal model. The second score results from modifying an existing score constructed for differential analysis of undirected networks. We develop strategies based on these scores to identify nodes responsible for topological differences between two Bayesian networks. Synthetic data and real-life data from designed experiments are used to demonstrate the efficacy of the proposed methods in detecting responsible nodes.

**KEYWORDS**

causality, designed experiment, interventional data, observational data, prediction invariance

## 1 | INTRODUCTION

Networks have been widely used to characterize interplays between components in a biological system. These components, referred to as nodes in a network, can be genes as in a gene regulatory network,[1] brain regions as in a brain network,[2] proteins as in a protein-protein interaction network,[3] or metabolites as in a metabolic network.[4] Comparing networks between different states can provide insight on mechanisms of disease initiation and progression. For example, changes in a gene regulatory network for a cancer patient compared with that for a healthy individual can lead to the detection of target genes necessary for developing targeted therapy; and contrasting the brain network of a cognitively impaired subject with that of a cognitively normal individual can help in discovering predictive biomarkers for neurodegenerative diseases. The analysis of differences between networks under different states is known as differential network analysis,[5,6] which has been an actively researched topic in a whole host of scientific domains besides biology.

Most existing works on differential network analysis consider comparisons between undirected networks. Shojaie[7] provides a comprehensive review on statistical methodologies for differential analysis of undirected networks. The common thread of these methods is to compare quantities that characterize marginal or conditional associations between nodes under different states. These quantities can be precision matrices, covariance matrices, or adjacency matrices corresponding to undirected networks under different states or associated with different populations, for example, a cancerous population and a healthy population. The ultimate goals of these analyses are, for instance, in the context of gene regulatory networks, to identify genes with differential gene expression levels, or to detect edges that suggest differential connections between genes.[8-12] More recently, Tu et al[13] incorporated information regarding whether or not genes are differentially expressed between two populations in their task of identifying differential edges.

Standing in stark contrast to the aforementioned works and references therein, we consider differential analysis of Gaussian Bayesian networks as the most widely applicable type of directed networks. Formulating a Bayesian network requires the specification of a directed acyclic graph (DAG) for a pre-specified set of nodes, along with the joint distribution of these nodes. For a Gaussian Bayesian network, the distribution of each node is assumed to be Gaussian marginally or conditioning on a linear combination of its parent nodes. A Bayesian network characterizes causal relationships between nodes, and thus captures richer information including and beyond associations and correlations between nodes. Research on differential analysis of Bayesian networks is still in its infancy. Recent developments in this direction mostly focus on proposing *direct methods* that bypass separately learning two Bayesian networks and directly estimate the so-called difference DAG. A difference DAG encapsulates information on changes in the existence of a causal effect, that is, the existence of a directed edge, as well as changes in the strengths of causal effects, that is, edge weights. In particular, Wang et al[14] developed an algorithm to infer the difference DAG by first testing invariance between regression coefficients that quantify causal effects, and then testing invariance in noise variances between the two models. Ghoshal et al[15] took the viewpoint of linear structural equation models for Bayesian networks so that estimating regression coefficients is equivalent to estimating a precision matrix. This allows them to leverage existing algorithms for computing the difference of precision matrix, and further estimate the difference DAG by repeatedly eliminating nodes and re-estimating the difference of precision matrix over the remaining nodes. These proposed strategies make use of data from observational studies under the assumption that two networks have a common topological order of nodes. A topological order of nodes compatible with a DAG is a linear ordering of nodes such that, corresponding to every directed edge in the DAG, a parent node comes before a child node in the ordering.

We propose in this study *indirect methods* for differential analysis of Bayesian networks based on data from designed experiments. Having data from designed experiments as opposed to data from observational studies allows us to infer causal relationships between nodes without assuming known ordering of nodes or two networks having the same topological order. Our methods are "indirect" in the sense that we do not directly infer the difference DAG; instead, we first infer two Bayesian networks separately, then we exploit the inferences to identify nodes responsible for potential topological discrepancies between two networks. It is also worth noting that, unlike many existing methods for differential network analysis (eg, Wang et al[14] and Ghoshal et al[15]) that aim to identify undirected or directed edges responsible for network differentiation, our methods identify responsible nodes. Section 2 provides a description of data available for drawing inference for Bayesian networks based on structural linear equation models. In Section 3 we define two new scores to quantify a node's potential to be a responsible node in differentiating two networks. Section 4 presents two methods for identifying responsible nodes based on the two new scores. Section 5 reports simulation studies where we apply the proposed and competing methods to synthetic data. Section 6 gives a close-up comparison of the two proposed methods and investigates uncertainty measures of a responsible node discovery based on these methods. We then use these methods to identify responsible nodes in a real-life application considered in Section 7. Section 8 summarizes contributions of the study and points out follow-up research directions.

## 2 | DATA AND MODELS

Consider $J$ nodes, $X_1, \dots, X_J$, that may exhibit different causal relationships in two populations of interest. For $\ell = 1, 2$, denote by $\mathbf{X}^{(\ell)}$ the $N_\ell \times J$ data matrix storing node-specific data of $N_\ell$ independent experimental units from population $\ell$. Experimental units in $\mathbf{X}^{(1)}$ may be independent or correlated with those in $\mathbf{X}^{(2)}$. For example, $X_1, \dots, X_J$ are $J$ genes that may interact with each other differently in a gene regulatory network associated with a disease population when compared with that for a control population; and a row in $\mathbf{X}^{(1)}$ records expression levels of $J$ genes from a healthy individual, whereas a row in $\mathbf{X}^{(2)}$ are expression levels of these genes from a cancer patient who is independent of individuals in the healthy control group. As an example with correlated experimental units under two states, in a brain network, $X_1, \dots, X_J$ are $J$ anatomic regions of interest (ROIs); the first row of $\mathbf{X}^{(1)}$ records the blood oxygen levels of these ROIs for an individual at the resting state, while the first row of $\mathbf{X}^{(2)}$ stores the blood oxygen levels of the ROIs for the same individual when performing a task. In this study, we assume that data $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are from a designed experiment involving sufficient interventions in various experimental conditions to allow identification of causality between nodes.

Denote by $\mathcal{E}$ the set of experimental conditions in the designed experiment. For node $X_j$, $j \in \{1, \dots, J\}$, its *observational data* are data observed under experimental conditions where $X_j$ is not intervened. We assume that the data generating mechanism of $X_j$'s observational data under condition $e \in \mathcal{E}$ is specified by the following structural linear equation model, for population $\ell \in \{1, 2\}$,

$$X_j = \sum_{k=1}^{J} \beta_{k,j}^{(\ell)} X_k + \epsilon_{\ell,j}^{(e)}, \text{ for } j = 1, \dots, J \tag{1}$$

where $\epsilon_{\ell,j}^{(e)}$ is the mean-zero Gaussian noise that is independent of all direct causal covariates of $X_j$, $\beta_{k,j}^{(\ell)}$ is the causal effect of $X_k$ on $X_j$ for $k \neq j$, and $\beta_{k,k}^{(\ell)} = 0$. It is assumed that data of each node are centered and thus an intercept is not needed in (1). In the nomenclature of Bayesian networks, if $\beta_{k,j}^{(\ell)} \neq 0$, then $X_k$ is a parent node of $X_j$. Denote by $\text{Pa}_j^{(\ell)}$ the parent set of $X_j$, and thus nodes in $\text{Pa}_j^{(\ell)}$ are direct causal nodes of $X_j$ under population $\ell$. Define $\mathbf{B}^{(\ell)} = [\beta_{k,j}^{(\ell)}]_{k,j=1,\dots,J}$ as the $J \times J$ causal effect coefficients matrix for population $\ell$, which determines the directed acyclic graph structure of population $\ell$, denoted by $G_\ell$. In Appendix A we use a concrete example to illustrate the data structure and models associated with a DAG outlined in this section.

# 3 TWO NEW SCORES FOR DIFFERENTIAL ANALYSIS

## 3.1 The prediction invariance score

An interesting property of causal models known as the *prediction invariance* property was discussed in detail in Peters et al,[16] and was utilized for causal inference. Put in the context of model (1), the prediction invariance property refers to the phenomenon that the model error when regressing $X_j$ on $\text{Pa}_j^{(\ell)}$ follows the same distribution across all experimental conditions where $X_j$ is not intervened. In contrast, when regressing $X_j$ on a proper subset of $\text{Pa}_j^{(\ell)}$, the model error may follow different distributions under different experimental conditions where $X_j$ is not intervened. That is, the prediction invariance property is not guaranteed to hold for an incorrectly specified causal model. In Appendix A we provide a concrete example that illustrates these statements by deriving the model error distributions of different regression models, some of which are causal models consistent with the true data generating process, but some are not.

Exploiting the prediction invariance property of a causal model, we construct a score that quantifies a node's potential to differentiate Bayesian networks by having different causal models under two states or populations. We call this score the prediction invariance score, PI for short. The following outlines the algorithm to compute PI scores of $J$ nodes given data $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, in which *Steps 4–6* are where we check if the prediction invariance property is violated under an assumed causal model based on $S$ partitions of $\mathcal{E}$.

*Step 1*: For $\ell = 1, 2$, estimate $\mathbf{B}^{(\ell)}$ based on $\mathbf{X}^{(\ell)}$ subject to the acyclic constraint. Denote by $\widehat{\mathbf{B}}^{(\ell)}(\widehat{G}_\ell) = [\widehat{\beta}_{k,j}^{(\ell)}(\widehat{G}_\ell)]_{k,j=1,\dots,J}$ the estimated coefficients matrix, and by $\widehat{G}_\ell$ the corresponding graph structure.

*Step 2*: Compute the least squares estimate of $\mathbf{B}^{(1)}$ using $\mathbf{X}^{(1)}$ while assuming the graph structure $\widehat{G}_2$, yielding the estimate denoted by $\widehat{\mathbf{B}}^{(1)}(\widehat{G}_2) = [\widehat{\beta}_{k,j}^{(1)}(\widehat{G}_2)]_{k,j=1,\dots,J}$. Similarly, obtain $\widehat{\mathbf{B}}^{(2)}(\widehat{G}_1) = [\widehat{\beta}_{k,j}^{(2)}(\widehat{G}_1)]_{k,j=1,\dots,J}$ using data $\mathbf{X}^{(2)}$ while assuming structure $\widehat{G}_1$.

For $j = 1, \dots, J$, implement Steps 3–7.

*Step 3*: Compute two residual vectors for node $X_j$ under population $\ell \in \{1, 2\}$,

$$\boldsymbol{r}_j^{(\ell)} = \mathbf{X}^{(\ell)}[O_j, j] - \sum_{k=1}^{J} \widehat{\beta}_{k,j}^{(\ell)}(\widehat{G}_\ell)\mathbf{X}^{(\ell)}[O_j, k], \tag{2}$$

$$\tilde{\boldsymbol{r}}_j^{(\ell)} = \mathbf{X}^{(\ell)}[O_j, j] - \sum_{k=1}^{J} \widehat{\beta}_{k,j}^{(\ell)}(\widehat{G}_m)\mathbf{X}^{(\ell)}[O_j, k], \text{ with } m \neq \ell, \tag{3}$$

where $O_j$ is the row index set of $\mathbf{X}^{(\ell)}$ corresponding to experimental units out of the $N_\ell$ units from population $\ell$ whose node $X_j$ is not intervened.

For $s = 1, \dots, S$, repeat Steps 4–6.

*Step 4*: Randomly divide $\boldsymbol{r}_j^{(\ell)}$ into two subsets of (nearly) equal size so that the two subsets correspond to two disjoint sets of experimental conditions. Denote these two subsets as $\boldsymbol{r}_{j,s,1}^{(\ell)}$ and $\boldsymbol{r}_{j,s,2}^{(\ell)}$, for $\ell = 1, 2$. Similarly, divide $\tilde{\boldsymbol{r}}_j^{(\ell)}$ into two subsets, $\tilde{\boldsymbol{r}}_{j,s,1}^{(\ell)}$ and $\tilde{\boldsymbol{r}}_{j,s,2}^{(\ell)}$, for $\ell = 1, 2$.

*Step 5*: For $\ell = 1, 2$, use an $F$ test to test the null hypothesis that $\boldsymbol{r}^{(\ell)}_{j,s,1}$ and $\boldsymbol{r}^{(\ell)}_{j,s,2}$ come from two populations that share the same variance; denote the $p$-value of the test as $v^{(\ell)}_{j,s}$. Carry out the same test on $\tilde{\boldsymbol{r}}^{(\ell)}_{j,s,1}$ and $\tilde{\boldsymbol{r}}^{(\ell)}_{j,s,2}$, denote the $p$-value of the test as $\tilde{v}^{(\ell)}_{j,s}$.

*Step 6*: For $\ell = 1, 2$, use a two-sample $t$ test to test the null hypothesis that $\boldsymbol{r}^{(\ell)}_{j,s,1}$ and $\boldsymbol{r}^{(\ell)}_{j,s,2}$ come from two populations that share the same mean; denote the $p$-value of the test as $m^{(\ell)}_{j,s}$. Carry out the same test on $\tilde{\boldsymbol{r}}^{(\ell)}_{j,s,1}$ and $\tilde{\boldsymbol{r}}^{(\ell)}_{j,s,2}$, denote the $p$-value of the test as $\tilde{m}^{(\ell)}_{j,s}$.

*Step 7*: Compute the PI score of $X_j$ given by

$$\text{PI}_j = \frac{1}{S}\sum_{s=1}^{S} \frac{\exp\left\{-\min\left(\tilde{m}^{(1)}_{j,s},\ \tilde{v}^{(1)}_{j,s},\ \tilde{m}^{(2)}_{j,s},\ \tilde{v}^{(2)}_{j,s}\right)\right\}}{\exp\left\{-\min\left(m^{(1)}_{j,s},\ v^{(1)}_{j,s},\ m^{(2)}_{j,s},\ v^{(2)}_{j,s}\right)\right\}}, \tag{4}$$

where $\min(a, b, c, d)$ refers to the minimum of $a, b, c, d$.

In Step 1, we apply the frequentist node-wise parent selection method proposed in our earlier work[17] to estimate a coefficients matrix that satisfies the acyclic constraint while encouraging sparsity. We then mismatch data with (estimated) graph structures when estimating causal effects again in Step 2 to obtain $\widehat{\mathbf{B}}^{(1)}(\widehat{G}_2)$ and $\widehat{\mathbf{B}}^{(2)}(\widehat{G}_1)$, which can (and usually do) differ from the coefficients matrix estimates in Step 1, that is, $\widehat{\mathbf{B}}^{(1)}(\widehat{G}_1)$ and $\widehat{\mathbf{B}}^{(2)}(\widehat{G}_2)$. If the two populations share the same causal model/structure for $X_j$ in the sense that $\text{Pa}^{(1)}_j = \text{Pa}^{(2)}_j$, then such mismatch does not lead to an incorrect causal model for $X_j$ under either population. Consequently, the residuals in both (2) and (3) from Step 3 are expected to satisfy the prediction invariance property, which in turn suggests that the $p$-values from Steps 5 and 6 should not be too small, despite how one splits the $|O_j|$ residuals into two subsets, and $\text{PI}_j$ in (4) is expected to be close to one. Here, $|O_j|$ denotes the cardinality of the index set $O_j$. Conversely, a value of $\text{PI}_j$ much larger than one implies violation of the prediction invariance under such mismatch, indicating that the causal model for $X_j$ under one population does not carry over to the other population in the sense that $\text{Pa}^{(1)}_j \neq \text{Pa}^{(2)}_j$. When carrying out a two-sample $t$ test in Step 6, we use the $t$ test involving a pooled sample variance if the corresponding equal-variance test in Step 5 fails to reject the null; otherwise, we use the $t$ test that assumes the Welch-Satterthwaite pooled degrees of freedom.[18] In Step 7, the denominator of the PI score in (4) serves as a normalization factor, and signifies that we quantify the severity of prediction-invariance violation after swapping the two graph structures relative to the severity before the swapping. Taking the exponential transformation in (4) is to avoid a nearly zero normalization factor at the denominator while keeping the transformation monotone; and choosing the smallest $p$-values to contrast between the numerator and denominator is to capture the most significant discrepancy between two sets of residuals revealed by the four equal-mean/variance tests.

We call $X_j$ a *differential node* if $\text{Pa}^{(1)}_j \neq \text{Pa}^{(2)}_j$, and a larger $\text{PI}_j$ can be interpreted as stronger data evidence indicating that $X_j$ is a differential node. In gene regulatory networks, a differential node is a gene that changes the way it is influenced by other genes as one moves from one state to the other, such as a mutated gene that acquires a new set of regulators as occurs frequently in cancer development.

## 3.2 | The DISCERN score and a modified score

With a similar goal as ours of identifying responsible nodes for differential network analysis, Grechkin et al[11] proposed a score to identify genes that are significantly rewired in the gene regulatory network of a disease population when comparing with a control population. Also like our methods, their score is constructed after undirected networks are inferred using a penalized estimation method to encourage sparse networks. They coin their method as DISCERN, standing for *differential sparse regulatory network*, and the score is called DISCERN score, defined as, for node $X_j, j \in \{1, \dots, J\}$,

$$\text{DISCERN}_j = \frac{\text{MSE}_j(1, 2) + \text{MSE}_j(2, 1)}{\text{MSE}_j(1, 1) + \text{MSE}_j(2, 2)}, \tag{5}$$

where

$$\mathrm{MSE}_j(\ell, m) = \frac{1}{|O_j|}\left\| \mathbf{X}^{(\ell)}[O_j, j] - \sum_{k=1}^{J} \hat{\beta}_{k,j}^{(m)}(\hat{G}_m)\mathbf{X}^{(\ell)}[O_j, k] \right\|^2, \text{ for } \ell, m \in \{1, 2\}, \tag{6}$$

is the mean squared error of predicting $X_j$ in population $\ell$ using the estimated covariate effects for population $m$, in which $||\boldsymbol{t}||$ denotes the $L_2$-norm of a vector $\boldsymbol{t}$. If $\ell = m$, the residual vector in (6) is $\boldsymbol{r}_j^{(\ell)}$ in (2). If $\ell \neq m$, the residual vector in (6) results from mismatching observational data of $X_j$ from one population with the covariate effects estimates for the other population. Due to this mismatch, a larger numerator in (5) in comparison to the denominator provides stronger data evidence suggesting that the association pattern and strength between $X_j$ and other nodes in one population do not carry over to the other population.

The DISCERN score shares some similarities with the PI score in that it also involves mismatching data from one population with coefficients estimates for the other population, and that the denominator in (5) also serves as a normalization factor like the denominator of PI. Despite these similarities, a responsible node identified by DISCERN should not be interpreted similarly as a differential node identified by PI. Nodes singled out by DISCERN are referred to as *perturbed nodes* in Grechkin et al,[11] and they contribute to changes in connectivity between nodes and edge weights in one undirected network when comparing with the other.

Adhering to our theme of causality-oriented differential analysis of Bayesian networks, we propose to revise $\mathrm{DISCERN}_j$ by replacing the mean squared error in (6) by

$$\mathrm{MSE}_j^*(\ell, m) = \frac{1}{|O_j|}\left\| \mathbf{X}^{(\ell)}[O_j, j] - \sum_{k=1}^{J} \hat{\beta}_{k,j}^{(\ell)}(\hat{G}_m)\mathbf{X}^{(\ell)}[O_j, k] \right\|^2, \text{ for } \ell, m \in \{1, 2\}. \tag{7}$$

Recall that $\{\hat{\beta}_{k,j}^{(\ell)}(\hat{G}_m),\ k = 1, \dots, J\}$ in (7) are entries of $\hat{\mathbf{B}}^{(\ell)}(\hat{G}_m)$ as an estimate of $\mathbf{B}^{(\ell)}$ using data $\mathbf{X}^{(\ell)}$ with the graph structure $\hat{G}_m$. Hence, when $\ell = m$, $\mathrm{MSE}_j^*(\ell, m)$ is the same as $\mathrm{MSE}_j(\ell, m)$ in (6); but when $\ell \neq m$, this new mean squared error depends on $\mathbf{X}^{(m)}$ only via $\hat{G}_m$, in contrast to the one defined in (6) that depends on $\mathbf{X}^{(m)}$ via $\hat{\mathbf{B}}^{(m)}(\hat{G}_m)$. In fact, the residual vector appearing in (7) is precisely $\tilde{\boldsymbol{r}}_j^{(\ell)}$ defined in (3), and it reflects prediction error of $X_j$ under population $\ell$ when one assumes that $\mathrm{Pa}_j^{(m)}$ contains all direct causal nodes of $X_j$. We call the modified score the **di**fferential **s**parse **c**ausal network (DISC) score, that is,

$$\mathrm{DISC}_j = \frac{\mathrm{MSE}_j^*(1, 2) + \mathrm{MSE}_j^*(2, 1)}{\mathrm{MSE}_j(1, 1) + \mathrm{MSE}_j(2, 2)}, \text{ for } j = 1, \dots, J, \tag{8}$$

which differs from $\mathrm{DISCERN}_j$ in (5) only in the numerator. A more substantial disagreement between the causal models for $X_j$ under two populations will lead to a larger numerator relative to the denominator of $\mathrm{DISC}_j$, and hence a node with a higher DISC score has a higher potential to be a differential node.

## 4 | RESPONSIBLE NODES IDENTIFICATION

The DISCERN, PI, and DISC scores all aim at identifying responsible nodes for network differentiation under two states/populations, with a higher value indicating a higher potential of being a responsible node. In a given application, it is unclear how high is enough for a score to support the claim of a responsible node, and the actual number of responsible nodes is typically unknown. Follow-up procedures are needed for picking out responsible nodes after a proposed score is computed for all $J$ nodes.

### 4.1 | Permutation-based method using DISCERN

Grechkin et al[11] assessed the significance of $\mathrm{DISCERN}_j$ by a $p$-value estimated via a permutation procedure. Applying this procedure to data from designed experiments entails permuting experimental units from the two populations under each experimental condition. The permuted versions of $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, denoted by $\mathbf{X}_b^{(1)}$ and $\mathbf{X}_b^{(2)}$, respectively, from the $b$-th permutation mimic two data sets from a common population, for $b = 1, \dots, M$. One then computes DISCERN scores for $J$ nodes based on data $\mathbf{X}_b^{(1)}$ and $\mathbf{X}_b^{(2)}$, starting from estimating $\mathbf{B}^{(\ell)}$ using data $\mathbf{X}_b^{(\ell)}$, for $\ell = 1, 2$. Denote the DISCERN score for $X_j$ from the $b$-th permutation by $\text{DISCERN}_{j,b}$, for $j = 1, \dots, J$ and $b = 1, \dots, M$. Then an estimated $p$-value associated with $\text{DISCERN}_j$ is given by $\sum_{b=1}^{M} I(\text{DISCERN}_{j,b} > \text{DISCERN}_j)/M$, where $I(\cdot)$ is the indicator function. After collecting $J$ estimated $p$-values, one implements the Benjamini-Hochberg false discovery rate correction[19] on these $p$-values to identify significant DISCERN scores while controlling the false discovery rate at a pre-specified level, such as 0.05. By the design of this permutation procedure, the resultant $p$-values are for testing the null hypothesis stating that $\mathbf{B}^{(1)} = \mathbf{B}^{(2)}$, that is, the two networks share the same structure and edge weights.

Our proposed scores $\text{PI}_j$ and $\text{DISC}_j$ are constructed to assess potential violation of a different null hypothesis, which states $\text{Pa}_j^{(1)} = \text{Pa}_j^{(2)}$. The above permutation procedure cannot be used or easily revised to assess the significance of $\text{PI}_j$ or $\text{DISC}_j$ because such permutation creates a null signifying that two Bayesian networks are identical in graph structure and also in the strength of causal effects. Had none of our scores for $J$ nodes been statistically significant, we would only conclude failing to reject the null hypothesis of $G_1 = G_2$, but cannot draw any conclusion regarding the hypothesis of $\mathbf{B}^{(1)} = \mathbf{B}^{(2)}$. It is unclear how to numerically create the null setting that only assumes two Bayesian networks share the same graph structure but may not have all identical edge weights via permutation or bootstrap. The assessment of statistical significance for the new scores remains an open question for future research.

## 4.2 | Rank-based methods using PI and DISC

To identify differential nodes based on the new scores, we propose a simple approach that utilizes the $p$-values from equal-mean and equal-variance tests for residuals $\{\tilde{\boldsymbol{r}}_j^{(1)}, \tilde{\boldsymbol{r}}_j^{(2)}, j = 1, \dots, J\}$ in (3). Because having small $p$-values among $\{\tilde{m}_{j,s}^{(1)}, \tilde{v}_{j,s}^{(1)}, \tilde{m}_{j,s}^{(2)}, \tilde{v}_{j,s}^{(2)}\}_{s=1}^{S}$ can already be evidence against the null $\text{Pa}_j^{(1)} = \text{Pa}_j^{(2)}$, these $p$-values shed light on the potential of $X_j$ being a differential node. We thus estimate the number of differential nodes to be

$$d = \left\lceil \frac{1}{S} \sum_{s=1}^{S} \max_{\ell \in \{1,2\}} \sum_{j=1}^{J} I\left( \min\left( \tilde{m}_{j,s}^{(\ell)}, \tilde{v}_{j,s}^{(\ell)} \right) < 0.025 \right) \right\rceil, \tag{9}$$

in which $\lceil t \rceil$ denotes the ceiling of a real number $t$. This estimator is to count, across $S$ random splits of $\tilde{\boldsymbol{r}}_j^{(\ell)}$, the average number of nodes for which a significant equal-mean test or a significant equal-variance test arises when we swap the graph structures for the two data sets. The cutoff value $0.025 (= 0.1/4)$ adopted in (9) results from applying the Bonferroni correction of multiple testing, acknowledging that there are four equal-mean/variance tests for each node, with each test set at a significance level of 0.1. Using a lower significance level, say, 0.05, for each test yields similar results in our empirical study. When PI scores are used, we choose $d$ nodes whose PI scores rank top $d$ to claim as differential nodes. Similarly, if DISC scores are used, we pick out $d$ nodes whose DISC scores rank top $d$ to claim as differential nodes.

Once a differential node is identified, we trace back to its parent sets in two networks. The discrepancy between two parent sets leads to additional nodes that drive the differentiation between two populations in the sense that they influence a differential node in one population but not in the other population. We call these nodes *driver nodes* as a second type of responsible nodes in addition to differential nodes. More specifically, suppose $X_j$ is a differential node, then a driver node $X_k \in \text{Pa}_j^{(1)} \Delta \text{Pa}_j^{(2)}$, where $\Delta$ is the symmetric difference operator. In gene regulatory networks, a driver node is a gene that changes how it influences or regulates other genes, for example, as one moves from one state/population to the other.

To provide an uncertainty measure for a claim of differential/driver node, we next formulate a likelihood assessment of a node being identified as such a responsible node according to a score. Let $\mathcal{D}$ and $\mathcal{V}$ be the set of differential nodes and the set of driver nodes, respectively, that reflect the ground truth, and denote by $\widehat{\mathcal{D}}$ and $\widehat{\mathcal{V}}$ the set of claimed differential nodes and the set of claimed driver nodes according to a score based on the estimated networks. We repeat the process of responsible nodes identification using a proposed score for $M$ bootstrap samples, each bootstrap sample created via sampling with replacement (within each experimental condition) from the raw data $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. The relative frequency of $X_j$ being concluded as a differential node across $M$ replicates is an empirical probability that $X_j$ is classified as a differential node given the current model and experimental settings, denoted by $P(X_j \in \widehat{\mathcal{D}})$. The empirical probability of $X_j$ being identified as a driver node is similarly computed, denoted by $P(X_j \in \widehat{\mathcal{V}})$.

# 5 | SIMULATION STUDY

## 5.1 | Competing methods

We now look into the performance of our proposed strategies for identifying responsible nodes in simulation studies where we generate data from designed experiments based on Bayesian networks we construct. Even though DISCERN is developed for identifying perturbed nodes in undirected networks, one may wonder what kind of nodes stand out according to DISCERN when applied to Bayesian networks. We thus include DISCERN as a competing method in the simulation and view perturbed nodes identified by it as differential nodes under the context of Bayesian networks. Another competing method is to simply claim a node to be differential if its parent sets are different in the two estimated DAGs, $\widehat{G}_1$ and $\widehat{G}_2$. This simple method does not provide information on the relative potential of identified nodes as responsible nodes, or uncertainty in the claims.

In summary, we compare four methods in the simulation study. A code name for each method is given following each Roman numeral label, (i) DAG: one picks out differential nodes by comparing $\widehat{G}_1$ and $\widehat{G}_2$; (ii) DISCERN: one claims nodes as differential nodes if their DISCERN scores are significant according to the permutation test; (iii) PI: one claims $d$ nodes as differential nodes whose PI scores rank top $d$; (iv) DISC: one claims $d$ nodes as differential nodes whose DISC scores rank top $d$. For the latter two methods, $d$ is computed according to (9). Following each of the four strategies of differential node identification, we pick out driver nodes accordingly for each method.

## 5.2 | Data generation

In the simulation study, we randomly generate two different DAGs of $J = 30$ nodes, $G_1$ and $G_2$, where we designate nodes as differential nodes, driver nodes, or neither. Appendix B describes a systematic approach to generate a pair of DAGs that allows users to specify the classification of nodes into $\mathcal{D}$, $\mathcal{V}$, or neither. After generating $G_1$ and $G_2$, we construct coefficients matrices, $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$, by filling in nonzero entries in them indicated by the corresponding graph structure, and each nonzero entry is generated from a two-component mixture, $0.5\ \mathcal{U}(-2,-1) + 0.5\ \mathcal{U}(1,2)$, where $\mathcal{U}(a,b)$ refers to a uniform distribution supported on $[a,b]$. Given $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$, we generate $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ from a designed experiment with various interventions to guarantee identifiability of causal relationships between nodes. More specifically, we follow the do interventions corresponding to the do operations in Pearl[20] to create $J$ experimental conditions, $\mathcal{E} = \{e_1, \dots, e_J\}$, so that, under $e_j$, $X_j$ (and only $X_j$) is intervened, with interventional data of $X_j$ generated independently of all other nodes, for $j = 1, \dots, J$. As a concrete example, we elaborate the generation of $\mathbf{X}^{(1)}$ next, starting with generating interventional data under $J$ experimental conditions. Under $e_j$, we use $n_j$ random numbers from $\mathcal{U}(0,1)$ as interventional data of $X_j$ in $\mathbf{X}^{(1)}$, for $j = 1, \dots, J$. After interventional data for each of $J$ nodes are generated, we generate observational data for each node, one node at a time following the topological order of $J$ nodes compatible with $G_1$, according to the model in (1) with $\epsilon_j^{(e_k)} \sim N(0, 0.25)$, where $k \neq j$, for $j = 1, \dots, J$.

In formulating $G_1$ and $G_2$, we create two cases in the simulation study that differ in the designation of responsible nodes besides other aspects. In Case 1, $G_1$ has 120 edges and $G_2$ has 111 edges, with $|\mathcal{D}| = 9$, $|\mathcal{V}| = 5$, and $|\mathcal{D} \cap \mathcal{V}| = 0$. In Case 2, $G_1$ has 196 edges and $G_2$ has 181 edges, with $|\mathcal{D}| = |\mathcal{V}| = 9$ and $|\mathcal{D} \cap \mathcal{V}| = 4$. When generating data under each case, we vary the number of experimental units in experimental condition $e_j$, that is, $n_j$, from 5 to 60, same for all $j = 1, \dots, J$. Under each simulation setting specified by a case number (1 or 2) and an $n_j$-level combination, we generate 100 Monte Carlo replicates of data matrices, $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. Four methods for differential node identification (DAG, DISCERN, PI, DISC) are applied to each pair of data matrices, followed by driver node identification. In the permutation procedure to assess the statistical significance of DISCERN scores, we set the number of permutations at $M = 300$. When computing PI scores, we randomly divide each residual vector, $\boldsymbol{r}_j^{(\ell)}$ and $\tilde{\boldsymbol{r}}_j^{(\ell)}$, into two halves ten times, that is, $S = 10$.

## 5.3 | Simulation results

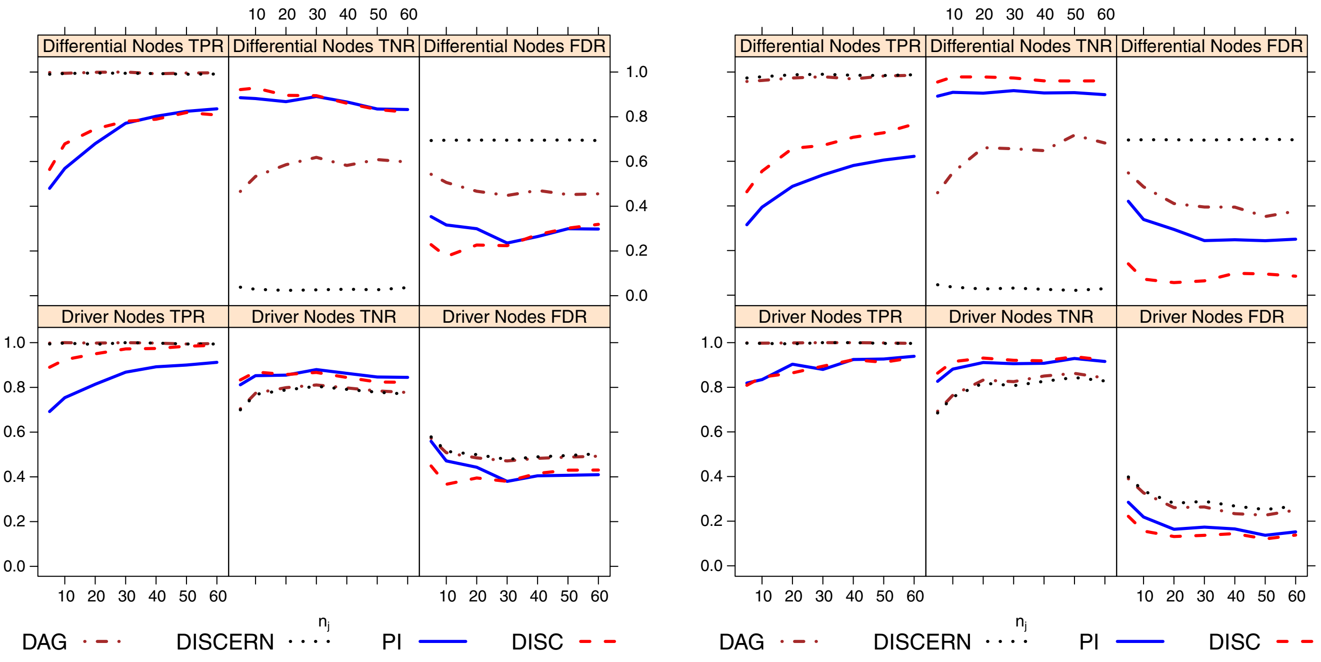
Three metrics are recorded in the simulation study to assess the performance of a method in terms of differential nodes identification: (a) the true positive rate (TPR) defined as the number of true differential nodes identified by the method divided by the actual number of differential nodes, that is, TPR $= |\widehat{\mathcal{D}} \cap \mathcal{D}|/|\mathcal{D}|$; (b) the true negative rate (TNR) defined as the number of true non-differential nodes concluded by the method divided by the actual number of non-differential nodes, that is, TNR $= |\widehat{\mathcal{D}}^c \cap \mathcal{D}^c|/|\mathcal{D}^c|$; (c) the false discovery rate (FDR) defined as the number of true non-differential nodes that are falsely claimed by the method as differential nodes divided by the total number of differential nodes claimed by the method, that is, FDR $= |\widehat{\mathcal{D}} \cap \mathcal{D}^c|/|\widehat{\mathcal{D}}|$. These metrics are similarly defined for driver nodes identification.

Figure 1a presents the Monte Carlo averages of each metric for four considered methods as $n_j$ varies under Case 1. These results suggest that PI and DISC perform similarly and satisfactorily in identifying differential nodes and driver nodes, especially when the sample size is not small. In comparison, DISCERN tends to claim a much larger number of differential nodes, which produces a TPR of almost one whilst the TNR is nearly zero despite the sample size, and its FDR is the highest among the four methods. Certainly, the method itself is not be blamed for its unsatisfactory performance since DISCERN is not developed for detecting violation of the null hypothesis stating that $G_1 = G_2$, but rather that $\mathbf{B}^{(1)} = \mathbf{B}^{(2)}$. Regardless, it is interesting that, with some modifications made to DISCERN, the new method DISC is much more effective in identifying differential nodes. The performance of DAG is similar to DISCERN in that it claims many more nodes to be differential nodes than PI and DISC, resulting in TPR nearly one, whereas its TNR and FDR lie between DISCERN and our two proposed methods.

Given $\widehat{G}_1$, $\widehat{G}_2$, and $\widehat{\mathcal{D}}$, identifying driver nodes is a deterministic process. In this regard, the four methods do not differ as drastically as in differential node identification, with our proposed methods outperforming DAG and DISCERN except when considering TPR. The two competing methods again yield almost perfect TPR, which is a direct consequence of their claiming majority of the nodes to be differential nodes in the first place. Results under Case 2 summarized in Figure 1b tell similar stories, but having some differential nodes that are also driver nodes in this case creates a more challenging scenario for PI and DISC to identify differential nodes, leading to lower TPR than those under Case 1.

Because it is preferable for a method to achieve high TPR and TNR while keeping FDR low, we define a ratio given by (TPR + TNR)/FDR to combine three metrics for assessing a method so that a higher ratio implies a better overall performance. Figure 2 shows the values of this ratio for different methods. From this angle, DISC stands out by giving the most satisfactory performance when all three metrics are considered via this ratio, PI substantially improves over DAG, and DISCERN is the least appealing method.

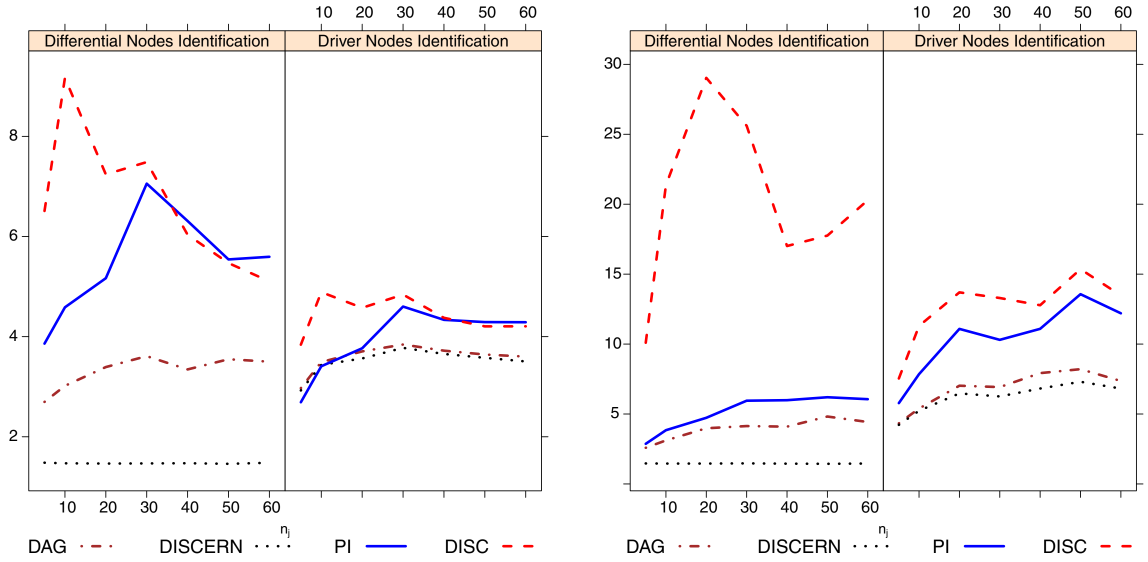
Additional simulation results are provided in Appendix C, where we consider a third configuration of $G_1$ and $G_2$. Following the graph generation strategy described in Appendix B, one has nearly full control of the roles different nodes



(A) Case 1

(B) Case 2

**FIGURE 1** Averages of TPR, TNR, and FDR across 100 Monte Carlo replicates in terms of differential nodes identification (upper panels) and driver nodes identification (lower panels) associated with four methods, DAG (dash-dotted lines), DISCERN (dotted lines), PI (sold lines), and DISC (dashed lines). (a) Case 1. (b) Case 2.



(A) Case 1

(B) Case 2

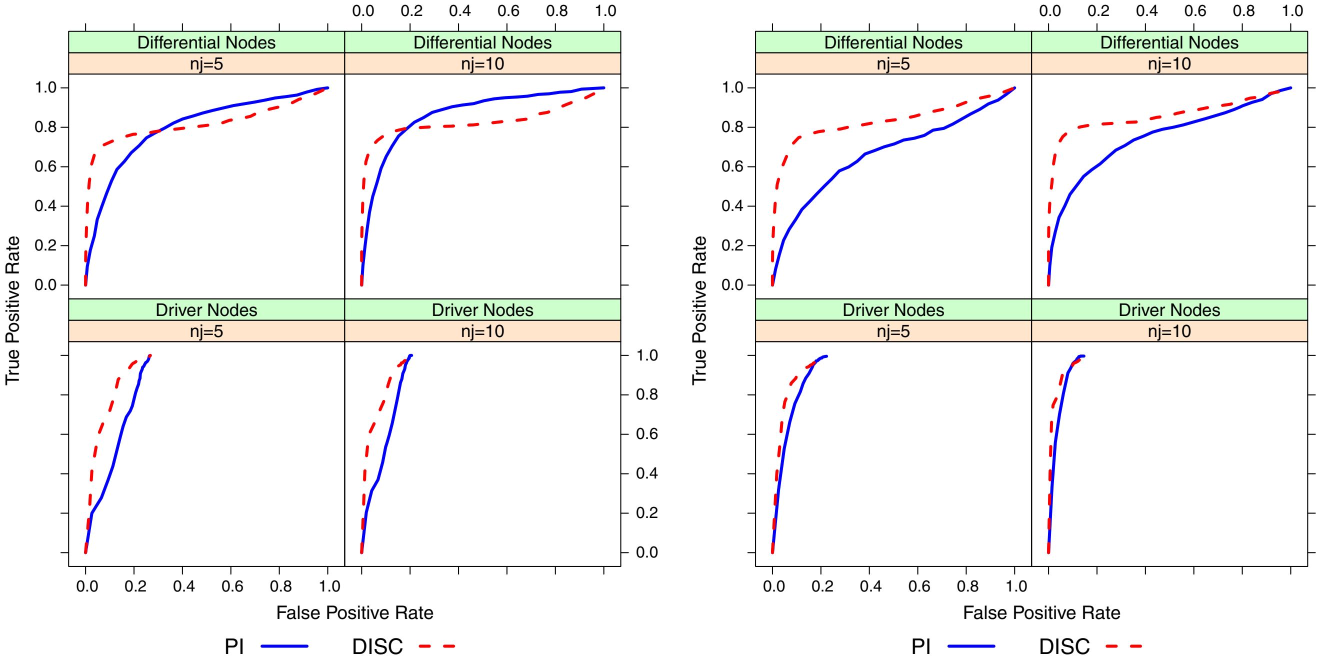
**FIGURE 2** Averages of the ratio, (TPR+TNR)/FDR, across 100 Monte Carlo replicates versus $n_j$ for differential nodes identification (left halves of (a) and (b)) and driver nodes identification (right halves of (a) and (b)) associated with four methods, DAG (dash-dotted lines), DISCERN (dotted lines), PI (sold lines), and DISC (dashed lines). (a) Case 1. (b) Case 2.

play when formulating two graph structures, as in Cases 1 and 2. This strategy also guarantees that the topological orderings of nodes according to $G_1$ and $G_2$ are compatible. The graph configuration presented in Appendix C breaks this pattern by having $G_1$ and $G_2$ that do not share compatible ordering of nodes.This additional setting highlights that we do not assume known ordering or a common ordering of nodes shared by two populations in our methodology development, which owes to the experimental data that allows for causal discovery, and the graph estimation method used in our methods that exploits such data for inferring causality. According to the empirical evidence presented in Appendix C, the comparisons between DAG, DISCERN, PI, and DISC when two populations have different ordering of nodes are similar to how these methods compare when two populations have the same orderings of nodes. Since all four strategies for responsible node identification start from estimating $G_1$ and $G_2$, one would expect that their performance depends on the choice of method for graph estimation. When the topological ordering of nodes is known or believed to be the same between two populations, different methods for graph estimation that make use of such known information can be adopted. Regardless, as long as causal relationships between nodes can be inferred (even partially) based on the available data using a chosen method for graph estimation, we expect that accounting for the inferred causality leads to more effective responsible node identification than when one ignores the causality information.

## 6 | FURTHER COMPARISON OF PI AND DISC

### 6.1 | Receiver operating characteristics

Having shown the efficacy of our two proposed methods in pinpointing responsible nodes for network differentiation, we now further compare their operating characteristics. The estimated number of differential nodes $d$ given in (9) utilizes information that go in the construction of PI scores. Using the so-defined $d$ as a threshold quantity to identify differential nodes based on both PI and DISC scores may appear to favor the former score. To alleviate the dependence on the choice of threshold when comparing PI and DISC in responsible nodes discovery, we inspect their receiver operating characteristics (ROC) as the threshold increases from 0 to $J$. Figure 3 presents ROC curves of PI and DISC regarding responsible nodes identification when they both claim top $k$ nodes as differential nodes based on the sorted scores, where $k \in \{0, 1, \dots, 30\}$, under the simulation settings formulated in Section 5.2 with $n_j \in \{5, 10\}$. According to these ROC curves that depict pairs of TPR and false positive rate (= $1 -$ TNR), we come to similar conclusions stated earlier: PI and DISC are comparable in



(A) Case 1

(B) Case 2

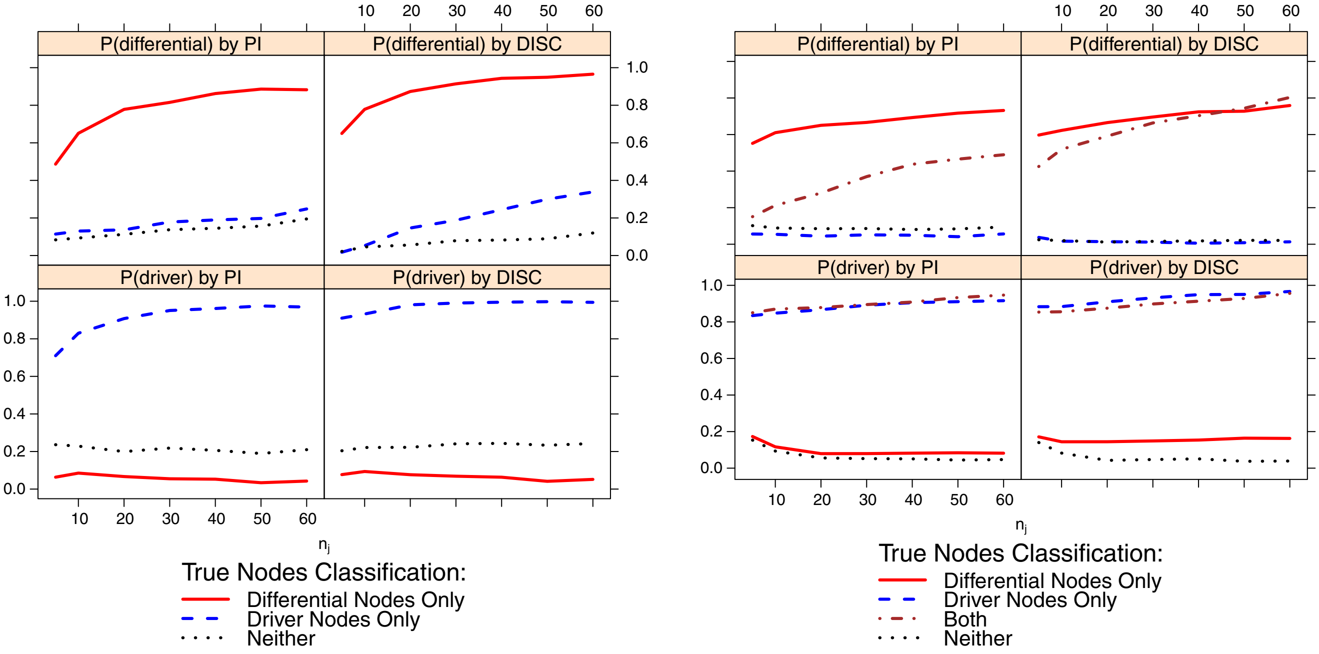
**FIGURE 3** Average receiver operating characteristic (ROC) curves of two proposed methods, PI (sold lines) and DISC (dashed lines), across 100 Monte Carlo replicates in terms of differential nodes identification (upper panels) and driver nodes identification (lower panels). (a) Case 1. (b) Case 2.

differential node identification when a responsible node does not play a dual role (as in Case 1), but DISC outperforms PI in identifying differential nodes when some of such nodes are also driver nodes (as in Case 2).

## 6.2 | Discriminating power

We now compare PI and DISC in regard to their ability to separate three types of nodes, which are differential nodes (in $\mathcal{D}$), driver nodes (in $\mathcal{V}$), and nodes that are neither (ie, in $\mathcal{D}^c \cap \mathcal{V}^c$). Let $\mathcal{A}$ generically refer to the set of nodes of a certain type that reflects the ground truth, such as $\mathcal{D}$, $\mathcal{V}$, or $\mathcal{D}^c \cap \mathcal{V}^c$; and let $\widehat{\mathcal{B}}$ be the set of nodes of a certain type that a method claims based on estimated networks. Define $\mathcal{P}(\widehat{\mathcal{B}}; \mathcal{A}) = |\mathcal{A}|^{-1} \sum_{X_j \in \mathcal{A}} P(X_j \in \widehat{\mathcal{B}})$ as the average empirical probability of a node being classified in $\widehat{\mathcal{B}}$ across all nodes that are actually in $\mathcal{A}$. For instance, $\mathcal{P}(\widehat{\mathcal{D}}; \mathcal{V}) = |\mathcal{V}|^{-1} \sum_{X_j \in \mathcal{V}} P(X_j \in \widehat{\mathcal{D}})$ is the average empirical probability of a driver node being classified as a differential node. Using simulated data $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ described in Section 5.2 under Case 1, we compute for each proposed method six averages of empirical probabilities: $\mathcal{P}(\widehat{\mathcal{D}}; \mathcal{A})$ and $\mathcal{P}(\widehat{\mathcal{V}}; \mathcal{A})$, for $\mathcal{A} = \mathcal{D},\ \mathcal{V}$, and $\mathcal{D}^c \cap \mathcal{V}^c$. Each empirical probability is obtained from $M = 300$ bootstrap samples. Figure 4a depicts Monte Carlo means of $\mathcal{P}(\widehat{\mathcal{D}}; \mathcal{A})$ and $\mathcal{P}(\widehat{\mathcal{V}}; \mathcal{A})$ across 100 replicates as $n_j$ varies for PI and DISC.

A method effective in identifying differential nodes is expected to yield $\mathcal{P}(\widehat{\mathcal{D}}; \mathcal{D})$ substantially higher than $\mathcal{P}(\widehat{\mathcal{D}}; \mathcal{A})$ for $\mathcal{A} \neq \mathcal{D}$, such as when $\mathcal{A} = \mathcal{V}$ or $\mathcal{A} = \mathcal{D}^c \cap \mathcal{V}^c$. Similarly, a method effective in identifying driver nodes should produce $\mathcal{P}(\widehat{\mathcal{V}}; \mathcal{V})$ much higher than $\mathcal{P}(\widehat{\mathcal{V}}; \mathcal{A})$ for $\mathcal{A} \neq \mathcal{V}$. These are indeed the comparative patterns demonstrated in Figure 4a for both proposed methods. Figure 4b provides a similar comparison under Case 2. Recall that, unlike in Case 1 where $|\mathcal{D} \cap \mathcal{V}| = 0$, here in Case 2 we have $|\mathcal{D} \cap \mathcal{V}| = 4$. We thus summarize in Figure 4b eight averages of empirical probabilities for each proposed method: $\mathcal{P}(\widehat{\mathcal{D}}; \mathcal{A})$ and $\mathcal{P}(\widehat{\mathcal{V}}; \mathcal{A})$, for $\mathcal{A} = \mathcal{D} \cap \mathcal{V}^c,\ \mathcal{D}^c \cap \mathcal{V},\ \mathcal{D} \cap \mathcal{V}$, and $\mathcal{D}^c \cap \mathcal{V}^c$. Overall, $\mathcal{P}(\widehat{\mathcal{D}}; \mathcal{A})$ still tends to be higher when $\mathcal{A}$ contains some differential nodes than when $\mathcal{A}$ excludes all differential nodes, although the power to correctly claim a differential node somewhat drops if this differential node is also a driver node, with the drop more noticeable for PI than for DISC. This suggests that DISC is less "confused" by the dual role of a responsible node than PI is. The two methods are both highly effective in separating driver nodes from non-driver nodes, whether or not a driver node is also a differential node.



(A) Case 1

(B) Case 2

**FIGURE 4** Monte Carlo means of $\mathcal{P}(\widehat{\mathcal{D}};\mathcal{A})$ (upper panels) and $\mathcal{P}(\widehat{\mathcal{V}};\mathcal{A})$ (lower panels) across 100 replicates associated with PI (left halves of (a) and (b)) and DISC (right halves of (a) and (b)) versus $n_j$, where $\mathcal{A}$ is $\mathcal{D}\cap\mathcal{V}^c$ (solid lines), $\mathcal{D}^c\cap\mathcal{V}$ (dashed lines), $\mathcal{D}\cap\mathcal{V}$ (dot-dashed lines), and $\mathcal{D}^c\cap\mathcal{V}^c$ (dotted lines), respectively. (a) Case 1. (b) Case 2.

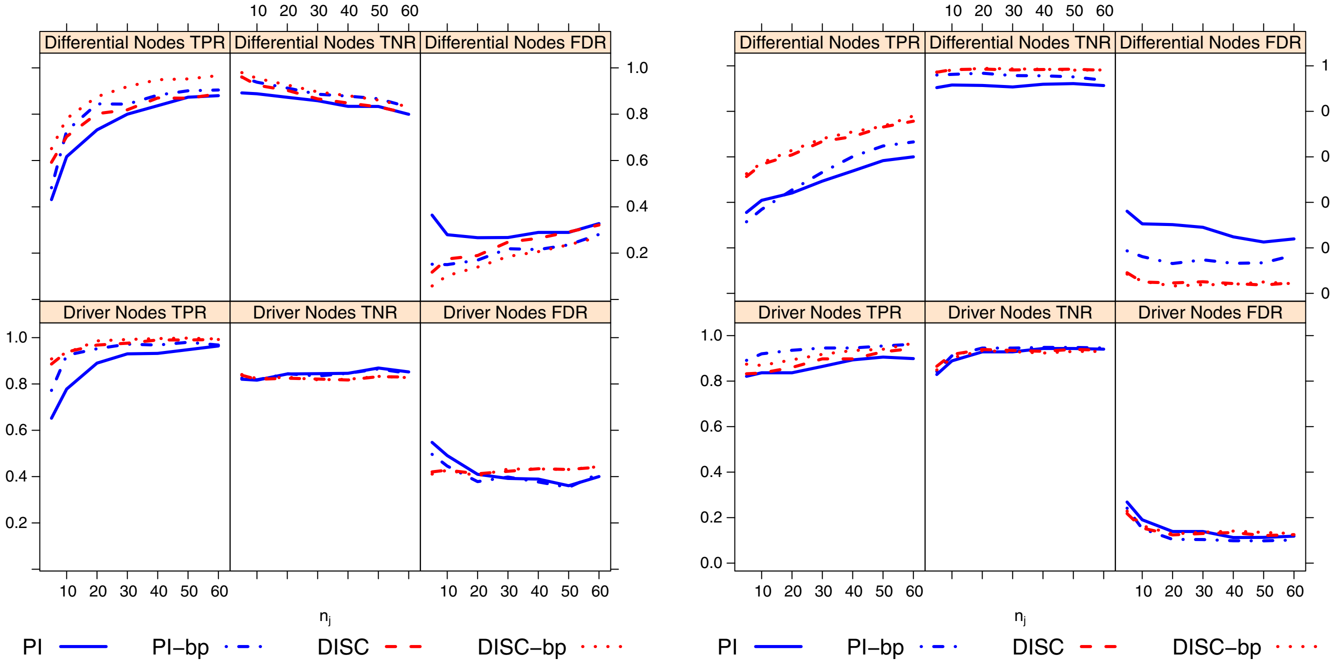
## 6.3 | Responsible nodes identification based on empirical probabilities

As evidenced in Figure 4, the empirical probabilities from bootstrap samples also provide clues for what role(s) a node potentially plays in Bayesian networks differentiation. In particular, $\{P(X_j \in \widehat{\mathcal{D}}), j = 1, \dots, J\}$ can separate differential nodes from non-differential nodes reasonably well, and $\{P(X_j \in \widehat{\mathcal{V}}), j = 1, \dots, J\}$ can distinguish driver nodes from non-driver nodes even better. Therefore, a simple strategy for identifying responsible nodes is to claim $X_j$ as a differential node if $P(X_j \in \widehat{\mathcal{D}}) > 0.5$, and to claim $X_j$ as a driver node if $P(X_j \in \widehat{\mathcal{V}}) > 0.5$. The threshold of 0.5 is an ad hoc choice before one has more theoretical ground to suggest a different threshold.

Under the same simulation settings described in Section 5.2, we demonstrate the operating characteristics of the new strategies of responsible nodes discovery based on the empirical probabilities, in comparison with our two previously proposed methods. To distinguish the new strategies from PI and DISC considered in Section 5 that do not involve empirical probabilities, we refer to the new strategy as PI-bp when PI scores are used to obtain the bootstrap-based probability, and as DISC-bp when DISC scores are used. Figure 5 shows the TPR, TNR, and FNR for responsible node identification associated with four methods: PI, PI-bp, DISC, and DISC-bp.

At the price of additional computation to obtain empirical probabilities, $\{P(X_j \in \widehat{\mathcal{D}})\}_{j=1}^J$ and $\{P(X_j \in \widehat{\mathcal{V}})\}_{j=1}^J$, one typically sees some improvement over the original method, PI or DISC. The improvement is more noticeable when PI scores are used, especially in lowering FDR of differential nodes. The gain is less impressive when DISC scores are used, especially under Case 2 where some nodes play dual roles. We thus conclude that the new strategies perform at least as well as the original proposed methods. Besides providing uncertain measures of claims regarding differential nodes, $\{P(X_j \in \widehat{\mathcal{D}})\}_{j=1}^J$ can also be viewed as scores of $J$ nodes that quantify the relative potential of being a differential node, with a higher probability indicating more potential under the current model and experimental settings. Similarly, $\{P(X_j \in \widehat{\mathcal{V}})\}_{j=1}^J$ can be used as $J$ scores to quantify the relative potential of these nodes as a driver node. Such scores can be more appealing due to their inherent probability interpretation that PI scores and DISC scores lack.

Lastly, we repeat the simulation study where we compare PI, PI-bp, DISC, and DISC-bp under Case 1, but we now generate random noise in each model in (1) from a mean-zero skew normal distribution[21] with variance 0.25 and skewness 0.47. This simulation study is designed to inspect the impact of violation of Gaussian Bayesian networks on the proposed



(A) Case 1

(B) Case 2

**FIGURE 5** Averages of TPR, TNR, and FDR across 100 Monte Carlo replicates in terms of differential nodes identification (upper panels) and driver nodes identification (lower panels) associated with four methods, PI (solid lines), PI-bp (dot-dashed lines), DISC (dashed lines), and DISC-bp (dotted lines). (a) Case 1. (b) Case 2.

methods. Figure 6 shows the three metrics for assessing the efficacy of each method in identifying responsible nodes. The deterioration of each considered method in the presence of non-Gaussian noise is evident when comparing with results in Figure 5a. The normality violation impacts the methods based on DISC scores more, and they are outperformed by the methods based on PI scores. This suggests that, even in the presence of model misspecification, the $t$ test and $F$ test, which contribute to the PI score, retain more information relating to causality discovery than the mean squared errors do, which are the building blocks of the DISC score. This makes PI scores more informative than DISC scores in the presence of model misspecification. Considering the empirical evidence in Section 5.3 and those in this section, we conclude that DISC is preferable to PI in responsible node identification unless when one has concern about the normality assumption for the model error, in which case we recommend opting for PI.

## 7 | APPLICATION TO FLOW CYTOMETRY DATA

In this section, we entertain the flow cytometry data collected from a designed experiment composed of nine experimental conditions described in Sachs et al,[22] where a series of stimulatory cues and inhibitory interventions were imposed on selected phosphorylated proteins and phospholipids. In this study, experimental units are human immune system cells, from each of which phosphomolecular measurements were collected from eleven phosphorylated proteins and phospholipids, viewed as nodes in a network.

For illustration purposes, we consider $J = 8$ of the eleven nodes, Raf, Mek, PLCg, PIP2, Erk, AKT, PKA, and PKC, that have both interventional data and observational data. We randomly select from the raw data a subset of size $N_1 = 123$ experimental units to form the first data matrix $\mathbf{X}^{(1)}$ corresponding to the eight nodes. We then make two changes in $\mathbf{X}^{(1)}$ to produce an artificial data set, $\mathbf{X}^{(2)}$, of the same size. First, we replace the observational data of PLCg in $\mathbf{X}^{(1)}$ with the median of these data plus standard normal random noise. Second, under the experimental condition where Mek was intervened, we substitute the interventional data of Mek with the median of Raf's data under this condition plus standard normal random noise. The first change is likely to make PLCg a differential node because the causal relationship between PLCg and its parent node(s) presumably supported by $\mathbf{X}^{(1)}$ is very likely to disappear after we distort the observational

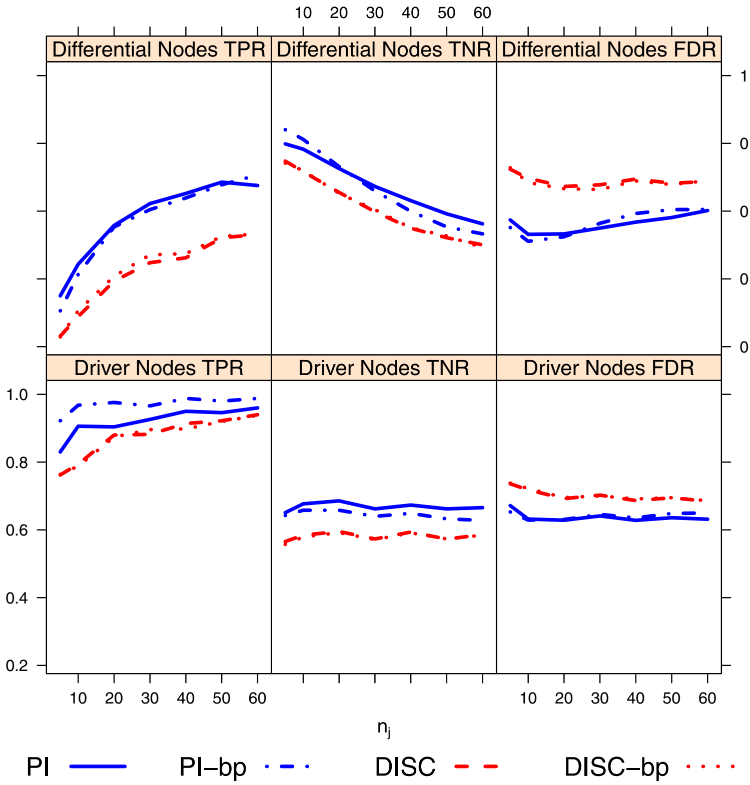


**FIGURE 6** Averages of TPR, TNP, and FDR across 100 Monte Carlo replicates under Case 1, with skewed normal random noise, in terms of differential nodes identification (upper panels) and driver nodes identification (lower panels) associated with four methods, PI (sold lines), PI-bp (dot-dashed lines), DISC (dashed lines), and DISC-bp (dotted lines).

data of PLCg to create $\mathbf{X}^{(2)}$. The second change is intended to make Mek a driver node by distorting its interventional data so that any potential bond between Mek and its child node is broken in the process of creating the artificial data.

Using $\mathbf{X}^{(1)}$ and the artificial data $\mathbf{X}^{(2)}$, we implement four methods, DAG, DISCERN, PI, and DISC, considered in Section 5 to find responsible nodes that contribute to differentiating two underlying populations. In practice, when it comes to PI and DISC for the purpose of finding responsible nodes, we recommend using DISC based on $d$ proposed in Section 4 instead of using empirical probabilities of node classification described in Section 6 whenever one is willing to assume Gaussian model error. In most practical settings, some nodes are very likely to be both differential nodes and driver nodes, which is the situation where DISC outperforms PI when the model error is Gaussian, and, according to evidence in Section 6.3, DISC gains little from using the empirical probabilities of node classifications. These empirical probabilities are more useful for uncertainty quantification after responsible nodes are claimed. Figure 7 gives the two estimated graphs, $\widehat{G}_1$ and $\widehat{G}_2$. Table 1 lists responsible nodes these methods claim. If only basing upon comparisons between $\widehat{G}_1$ and $\widehat{G}_2$, one would claim almost all nodes as differential nodes. In this particular application, DISCERN becomes less aggressive in making claims about differential nodes than PI and DISC; in particular, it does not pick out PLCg as a differential node. Our proposed methods, PI and DISC, are mostly in agreement in differential nodes identification, and both give high confidence in the claim that PLCg is a differential node. They also agree on driver nodes identification, with high empirical probabilities assigned to Mek.

## 8 | DISCUSSION

We defined two scores, the PI score and DISC score, for identifying responsible nodes that contribute to differentiating the Bayesian network under two states or associated with two populations. Both scores are designed to collect data evidence against the null $G_1 = G_2$, with the scores for node $X_j$ tailored for testing the null $\mathrm{Pa}_j^{(1)} = \mathrm{Pa}_j^{(2)}$. Using intermediate results needed for computing these scores, we proposed an estimator for the total number of differential nodes to facilitate differential node identification based on the ranked scores. Lastly, we employed a bootstrap procedure to obtain uncertainty

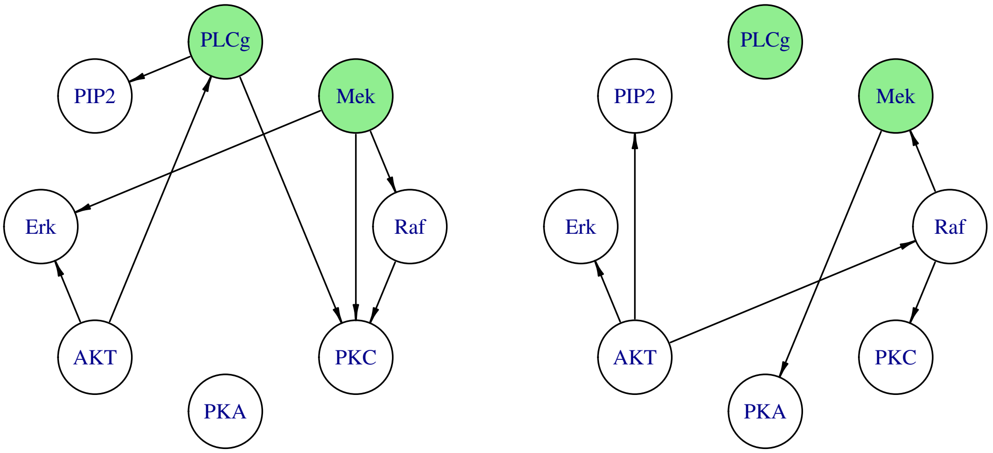


**FIGURE 7** The estimated directed graphs based on a subset of the flow cytometry data set $\mathbf{X}^{(1)}$ (on the left) and the one based on an artificial data set $\mathbf{X}^{(2)}$ (on the right). Some of the data in $\mathbf{X}^{(1)}$ associated with PLCg and Mek are distorted to produce $\mathbf{X}^{(2)}$.

**TABLE 1** Responsible nodes identification associated with each of four methods using the flow cytometry data set $\mathbf{X}^{(1)}$ and an artificial data set $\mathbf{X}^{(2)}$.

| | Differential nodes |
|---|---|
| DAG | All nodes except for AKT |
| DISCERN | PKC (0.000), PIP2 (0.007) |
| PI | PLCg (1.000), Raf (1.000), PKA (0.913), PIP2 (0.630) |
| DISC | PLCg (1.000), Raf (1.000), PKC (1.000), PIP2 (0.997) |
| | Driver nodes |
| DAG | Mek, AKT, PLCg, Raf |
| DISCRERN | Mek, AKT, PLCg |
| PI | Mek (1.000), AKT (1.000), PLCg (0.903) |
| DISC | Mek (1.000), AKT (1.000), PLCg (1.000) |

*Note*: Numbers in parentheses for DISCERN are *p*-values for the significance of chosen nodes based on 300 permutations. Numbers in parentheses for PI and DISC are empirical probabilities based on 300 bootstrap samples for uncertainty measures of claimed nodes.

assessments of claims regarding differential/driver node discovery in the form of empirical probabilities. These empirical probabilities can also be used as scores to quantify nodes' potential of being a certain type of responsible nodes. Computer programs for implementing the proposed methods are available at https://github.com/hxzusc/DiffNet.

Compared with the method of DISCERN developed for undirected networks, and with the simple method based on inspecting structural discrepancies between two estimated DAG's, the proposed methods based on PI and DISC scores lead to more accurate discoveries of responsible nodes. Even though the new scores do not aim to detect discrepancies in the magnitude of causal effects between two populations, they make use of such information indirectly via their dependence on different forms of residuals. These residuals are the key to revealing different causal structures of two Bayesian networks. As a referee pointed out, to test whether or not residuals from different experimental conditions follow the same distribution, one may consider other tests, such as the Kolmogorov–Smirnov test. We use the *t* test and *F* test here to focus on comparing the mean and variance of residuals under different conditions for simplicity, which is also well-motivated when residuals are viewed as Gaussian errors.

Implementation of both proposed methods starts with inferring two Bayesian networks. This is also the most computationally heavy step of the proposed methods. Future improvements on the computer programs include implementing parallel computing to shorten the computation time. With a large collection of existing methods for estimating Bayesian networks based on data from designed experiments, properties of the proposed methods when using different approaches to infer Bayesian networks are worthy of systematic investigation. In particular, if data associated with the two networks are believed to be dependent, a method that accounts for such dependence may be more efficient in responsible nodes identification than our current methods that ignore such information. Another important topic for follow-up research is the nature of responsible nodes singled out by these methods when causal relationships are only partially identifiable due to lack of rich enough experimental data.

## ACKNOWLEDGEMENTS

We are grateful to the two anonymous referees and the Associate Editor, whose comments and feedback on our work help us improve the manuscript substantially. Zhang's work is supported by National Institutes of Health, NIAID grant R01AI121226.

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## ORCID

*Xianzheng Huang* https://orcid.org/0000-0001-7077-0869
*Hongmei Zhang* https://orcid.org/0000-0003-3557-0364

## REFERENCES

1. Emmert-Streib F, Dehmer M, Haibe-Kains B. Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Front Cell Develop Biol*. 2014;2:38.
2. Bullmore E, Sporns O. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nat Rev Neurosci*. 2009;10(3):186-198.
3. Hao T, Peng W, Wang Q, Wang B, Sun J. Reconstruction and application of protein–protein interaction network. *Int J Mol Sci*. 2016;17(6):907.
4. Christensen B, Nielsen J. Metabolic network analysis: a powerful tool in metabolic engineering. *Bioanal Biosensors Bioprocess Monitor*. 2000;66:209-231.
5. Ideker T, Krogan NJ. Differential network biology. *Mol Syst Biol*. 2012;8(1):565.
6. Ruan D, Young A, Montana G. Differential analysis of biological networks. *BMC Bioinform*. 2015;16(1):1-13.
7. Shojaie A. Differential network analysis: a statistical perspective. *Wiley Interdiscip Rev: Comput Stat*. 2021;13(2):e1508.
8. Amar D, Safer H, Shamir R. Dissection of regulatory networks that are altered in disease via differential co-expression. *PLoS Comput Biol*. 2013;9(3):e1002955.
9. Bockmayr M, Klauschen F, Györffy B, Denkert C, Budczies J. New network topology approaches reveal differential correlation patterns in breast cancer. *BMC Syst Biol*. 2013;7(1):1-14.
10. Ha MJ, Baladandayuthapani V, Do KA. DINGO: differential network analysis in genomics. *Bioinformatics*. 2015;31(21):3413-3420.
11. Grechkin M, Logsdon BA, Gentles AJ, Lee SI. Identifying network perturbation in cancer. *PLoS Comput Biol*. 2016;12(5):e1004888.
12. Xie J, Yang F, Wang J, et al. DNF: a differential network flow method to identify rewiring drivers for gene regulatory networks. *Neurocomputing*. 2020;410:202-210.
13. Tu JJ, Ou-Yang L, Zhu Y, Yan H, Qin H, Zhang XF. Differential network analysis by simultaneously considering changes in gene interactions and gene expression. *Bioinformatics*. 2021;37(23):4414-4423.
14. Wang Y, Squires C, Belyaeva A, Uhler C. Direct estimation of differences in causal graphs. *Adv Neural Inf Process Syst*. 2018;31:3770-3781.
15. Ghoshal A, Bello K, Honorio J. Direct learning with guarantees of the difference dag between structural equation models. arXiv preprint arXiv:1906.12024 2019.
16. Peters J, Bühlmann P, Meinshausen N. Causal inference by using invariant prediction: identification and confidence intervals. *J R Stat Soc Stat Methodol Series B*. 2016;78(5):947-1012.
17. Huang X, Zhang H. Corrected score methods for estimating Bayesian networks with error-prone nodes. *Stat Med*. 2021;40(11):2692-2712.
18. Welch BL. The generalization of "STUDENT'S"2 problem when several different population varlances are involved. *Biometrika*. 1947;34(1-2):28-35.
19. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc B Methodol*. 1995;57(1):289-300.
20. Pearl J. Causal inference in statistics: an overview. *Stat Surv*. 2009;3:96-146.
21. Azzalini A. A class of distributions which includes the normal ones. *Scand J Stat*. 1985;12:171-178.
22. Sachs K, Perez O, Pe'er D, Lauffenburger DA, Nolan GP. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*. 2005;308(5721):523-529.

**How to cite this article:** Huang X, Zhang H. Detecting responsible nodes in differential Bayesian networks. *Statistics in Medicine*. 2024;43(17):3294-3312. doi: 10.1002/sim.10125

## APPENDIX A. AN ILLUSTRATIVE EXAMPLE FOR PREDICTION INVARIANCE

As an illustrative example, we consider four nodes, $X_1$, $X_2$, $X_3$, and $X_4$, whose causal relationships had there been no intervention are specified by the DAG in Figure A1. Using notations introduced in Section 2 but without the population index $\ell$, we use $\mathbf{X}$ to denote an $N \times 4$ data matrix that contains data of the four nodes, where the first $N/2$ rows of data are from the first experimental condition $e_1$ where $X_2$ is suppressed at zero, and the latter $N/2$ rows of $\mathbf{X}$ are from the second condition $e_2$ under which $X_3$ is stimulated so that it follows some non-Gaussian distribution. Knowing that Figure A1 depicts the causal relationships of the four nodes, we have the structural linear equation models for $X_1$ and $X_4$ given by, for $e \in \{e_1, e_2\}$ since $X_1$ and $X_4$ are not intervened under these conditions,

$$X_1 = \beta_{2,1}X_2 + \beta_{3,1}X_3 + \epsilon_1^{(e)}, \tag{A1}$$

$$X_4 = \beta_{1,4}X_1 + \epsilon_4^{(e)}, \tag{A2}$$

where $\epsilon_1^{(e)} \perp (X_2, X_3)$ and $\epsilon_4^{(e)} \perp X_1$ are mean-zero Gaussian noise. In other words, the first column of $\mathbf{X}$, $\mathbf{X}[,1]$, is the observational data for $X_1$ that arises from (A1); similarly, $\mathbf{X}[,4]$ contains observational data for $X_4$ arising from (A2). Due to the intervention in $e_1$, entries in $\mathbf{X}[1 : N/2,\ 2]$ are all zero's, and data in $\mathbf{X}[(N/2+1) : N,\ 2]$ contains observational data of $X_2$ coming from the model $X_2 = \epsilon_2^{(e_2)}$, where $\epsilon_2^{(e_2)}$ is mean-zero Gaussian noise since $X_2$ is a root node according to Figure A1. Lastly, under $e_1$ where $X_3$ is not intervened, its distribution conditional on its only parent $X_2$ is specified by $X_3 = \beta_{2,3}X_2 + \epsilon_3^{(e_1)}$, where $\epsilon_3^{(e_1)} \perp X_2$ is mean-zero Gaussian noise. The aforementioned covariate effects coefficients, $\beta_{2,1}$, $\beta_{3,1}$, $\beta_{1,4}$ and $\beta_{2,3}$, are the only non-zero entries in the coefficients matrix $\mathbf{B} = [\beta_{k,j}]_{k,j=1,2,3,4}$.

Next we explain the prediction invariance property of causal models not limited to linear models with Gaussian error. The graph in Figure A1 leads to a factorization of the joint distribution of $(X_1, X_2, X_3, X_4)$ elicited by the following hierarchical models,

$$X_2 \sim f_2(x_2), \tag{A3}$$

$$X_3 | X_2 \sim f_{3|2}(x_3|x_2), \tag{A4}$$

$$X_1 | (X_2, X_3) \sim f_{1|(2,3)}(x_1|x_2, x_3), \tag{A5}$$

$$X_4 | X_1 \sim f_{4|1}(x_4|x_1), \tag{A6}$$

where the probability density function (pdf) of a distribution is used to specify a distribution in each of the four submodels. For example, $f_{1|(2,3)}(x_1|x_2, x_3)$ can be the pdf of $N(\beta_{2,1}X_2 + \beta_{3,1}X_3, 1)$, indicating that $X_1|(X_2, X_3) \sim N(\beta_{2,1}X_2 + \beta_{3,1}X_3, 1)$. Note that, as long as this conditional distribution remains the same, not in terms of the specific conditional mean or variance but in terms of the distribution family (Gaussian in this example), the model error when regressing $X_1$ on $(X_2, X_3)$, that is, $\epsilon = X_1 - (\beta_{2,1}X_2 + \beta_{3,1}X_3)$, follows the same distribution, $N(0, 1)$ in this example, despite what values $X_2$ and $X_3$ are evaluated at. Hence, to check whether or not the model error of a structural equation model follows the same distribution

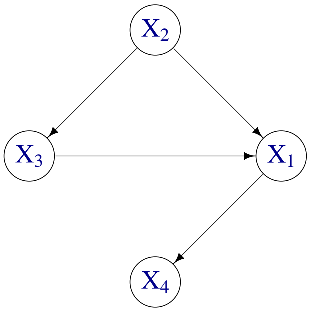


FIGURE A1 A directed acyclic graph with four nodes for the causal relationships between them had there been no interventions.

across different experimental conditions in a designed experiment, it suffices to check whether or not the conditional distribution of the response given covariates in the structural equation model stays in the same distribution family across different experimental conditions.

Now consider four structural equation models for $X_1$, referred to as models A, B, C, and D, with the set of covariates being $\{X_2\}$, $\{X_2, X_4\}$, $\{X_2, X_3\}$, and $\{X_2, X_3, X_4\}$, respectively. One fits each model to two data sets successively, collected under two experimental conditions in $\mathcal{E} = \{e_1, e_2\}$. Under $e_1$, $X_2$ is intervened so that $X_2 \sim f_2^*(x_2) \neq f_2(x_2)$. Under $e_2$, $X_3$ is intervened so that $X_3$ is independent of all other nodes, leading to a model replacing (A4) under this experimental condition, $X_3 \sim f_3^*(x_3) \neq f_{3|2}(x_3|x_2)$. In effect, the intervention in $e_1$ only impacts (A3) in the above hierarchical models, and it does not change the graph structure in Figure A1, whereas the intervention under $e_2$ erases the edge connecting $X_2$ and $X_3$. We next inspect whether or not the model error distribution associated with each of models A, B, C, and D changes when one moves from experimental condition $e_1$ to experimental condition $e_2$. As we point out earlier, this is equivalent to inspecting if the distribution of $X_1$ given the set of covariates in a structural equation model remains the same (in terms of distribution family) under the two experimental conditions. According to Figure A1, the set of direct causal nodes of $X_1$ is $\text{Pa}_1 = \{X_2, X_3\}$. Since the set of covariates in model C coincides with $\text{Pa}_1$, model C is a correct causal model for $X_1$. Clearly, the conditional distribution of $X_1$ given $\{X_2, X_3\}$ remains the same whether $X_2$ or $X_3$ is intervened. Thus the model error of model C remains the same under $e_1$ and $e_2$, that is, prediction invariance holds for model C. In contrast, for model A, we have the conditional distribution of the response $X_1$ given the covariate $X_2$ specified by

$$\begin{aligned} f_{1|2}^{(1)}(x_1|x_2) &= \int f_{1|2,3}(x_1|x_2, v) f_{3|2}(v|x_2)dv, \text{ under } e_1, \\ f_{1|2}^{(2)}(x_1|x_2) &= \int f_{1|2,3}(x_1|x_2, v) f_3^*(v)dv, \text{ under } e_2, \end{aligned} \tag{A7}$$

where the integrations are over the support of $X_3$. By suppressing the causal dependence of $X_3$ on $X_2$ under $e_2$, one ends up with a model error distribution differs from that under $e_1$ in general according to (A7). This is an example where one loses prediction invariance due to missing a direct causal covariate ($X_3$) in a regression model. In other words, model A is an incorrect causal model for $X_1$. For model B, the conditional distribution of the response given $(X_2, X_4)$ under condition $e_k$ is specified by, for $k = 1, 2$,

$$f_{1|(2,4)}^{(k)}(x_1|x_2, x_4) = \frac{f_{4|1}(x_4|x_1) f_{1|2}^{(k)}(x_1|x_2)}{\int f_{4|1}(x_4|v) f_{1|2}^{(k)}(v|x_2)dv},$$

where the integration is over the support of $X_1$. Because $f_{1|2}^{(1)}(x_1|x_2) \neq f_{1|2}^{(2)}(x_1|x_2)$ in general by (A7), the above conditional distribution can differ between the two experimental conditions. Hence, model B does not possess the prediction invariance property either, which is another incorrect causal model for $X_1$. Lastly, the model error distribution of model D remains the same under $e_1$ and $e_2$ because

$$f_{1|(2,3,4)}(x_1|x_2, x_3, x_4) = \frac{f_{1|(2,3)}(x_1|x_2, x_3) f_{4|1}(x_4|x_1)}{\int f_{1|(2,3)}(v|x_2, x_3) f_{4|1}(x_4|v)dv},$$

where neither the distribution of $X_1$ conditional on $(X_2, X_3)$ nor the distribution of $X_4$ conditional on $X_1$ is affected by the interventions imposed under $e_1$ and $e_2$. Model D serves as another example under which prediction invariance holds because, like model C, it is also a correct causal model for $X_1$, although it is less parsimonious than model C.

## APPENDIX B. DESIGN OF TWO DIRECTED ACYCLIC GRAPHS

In order to control which nodes are differential nodes, driver nodes, or neither when constructing two DAG's, we first randomly generate $G_1$, and then strategically revise entries in the adjacency matrix corresponding to $G_1$ to create a different adjacency matrix that specifies $G_2$. With an abuse of notation, we also use $G_\ell$ to denote the adjacency matrix corresponding to graph $G_\ell$.

**TABLE B1** Partition of a $J \times J$ adjacency matrix in order to design $G_2$ that differs from $G_1$ systematically, with user-designated differential nodes and driver nodes.

| | **1** | **…** | ***a*** | ***a* + 1** | **…** | ***a* + *b*** | ***a* + *b* + 1** | **…** | ***J*** |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | |
| ⋮ | | | | | | | | | |
| $a$ | | | | | | | | | |
| $a+1$ | | | | | | | | | |
| ⋮ | | | | Block 1 | | | Block 2 | | |
| $a+b$ | | | | | | | | | |
| $a+b+1$ | | | | | | | | | |
| ⋮ | | | | Block 3 | | | Block 4 | | |
| $J$ | | | | | | | | | |

*Note*: The gray region corresponds to Block 5.

To construct $G_2$, we partition its adjacency matrix into five blocks as shown in Table B1, where we partition $J$ nodes into three groups, $\{X_1, \dots, X_a\}$, $\{X_{a+1}, \dots, X_{a+b}\}$, and $\{X_{a+b+1}, \dots, X_J\}$. The row name and column name correspond to the row index and column index of the adjacency matrix in Table B1.

Following this partition of the adjacency matrix, given $G_1$, we create $G_2$ by keeping the same block 5 as that in $G_1$ to make nodes in $\{X_1, \dots, X_a\}$ neither differential nodes nor driver nodes. We then randomly replace nonzero entries in one or multiple blocks among blocks 1–4 of $G_1$ with zeros to lead to $G_2$. This amounts to random edge deletion in $G_1$ to produce $G_2$, which guarantees acyclic constraint satisfied in $G_2$ as long as $G_1$ is a directed acyclic graph. In the simulation study presented in the main article, we generate two pairs of graphs referred to as Case 1 and Case 2 as follows.

- Case 1: Keep blocks 1, 2, and 4 the same as those in $G_1$, and randomly replace nonzero entries in block 3 of $G_1$ with zeros to create block 3 in $G_2$. In this case, some nodes in $\{X_{a+1}, \dots, X_{a+b}\}$ become differential nodes but not driver nodes, and some nodes in $\{X_{a+b+1}, \dots, X_J\}$ are driver nodes but not differential nodes.
- Case 2: Keep blocks 2 and 4 the same as those in $G_1$, and randomly replace nonzero entries in blocks 1 and 3 of $G_1$ with zeros to create blocks 1 and 3 in $G_2$. By so doing, some nodes in $\{X_{a+1}, \dots, X_{a+b}\}$ are differential nodes and also driver nodes, and some nodes in $\{X_{a+b+1}, \dots, X_J\}$ are driver nodes but not differential nodes.

## APPENDIX C. ADDITIONAL SIMULATION RESULTS

Besides the two cases of formulating $G_1$ and $G_2$ as described in Section 5.2, we consider a third specification of $G_1$ and $G_2$ that does not follow the designs in Appendix B. More specifically, after generating $G_1$ with $p = 30$ and 120 edges, we randomly permute some columns of the adjacency matrix of $G_1$, followed by deleting cycles of the permuted adjacency matrix to obtain the final adjacency matrix that specifies $G_2$ with 117 edges. The resultant graph $G_2$ is incompatible with any topological ordering of nodes that $G_1$ is compatible with. This design of $G_1$ and $G_2$ produces $|\mathcal{D}| = 8$, $|\mathcal{V}| = 7$, and $|\mathcal{D} \cap \mathcal{V}| = 1$. Figure C1 summarizes simulation results from this configuration of the two graphs, with data simulated following the rest of the simulation designs in Section 5.2.

Most patterns of the comparisons between the four considered methods in this case are similar to those under Case 2 (see Figures 1 and 2). In particular, the two proposed methods, PI and DISC, are more effective in identifying differential nodes than the method based on the DISCERN score, which is inferior to the simple method based only on the two estimated DAG's. Between the two proposed methods, DISC outperforms PI, especially when the sample size is small. The four considered methods are comparable in terms of driver nodes identification.

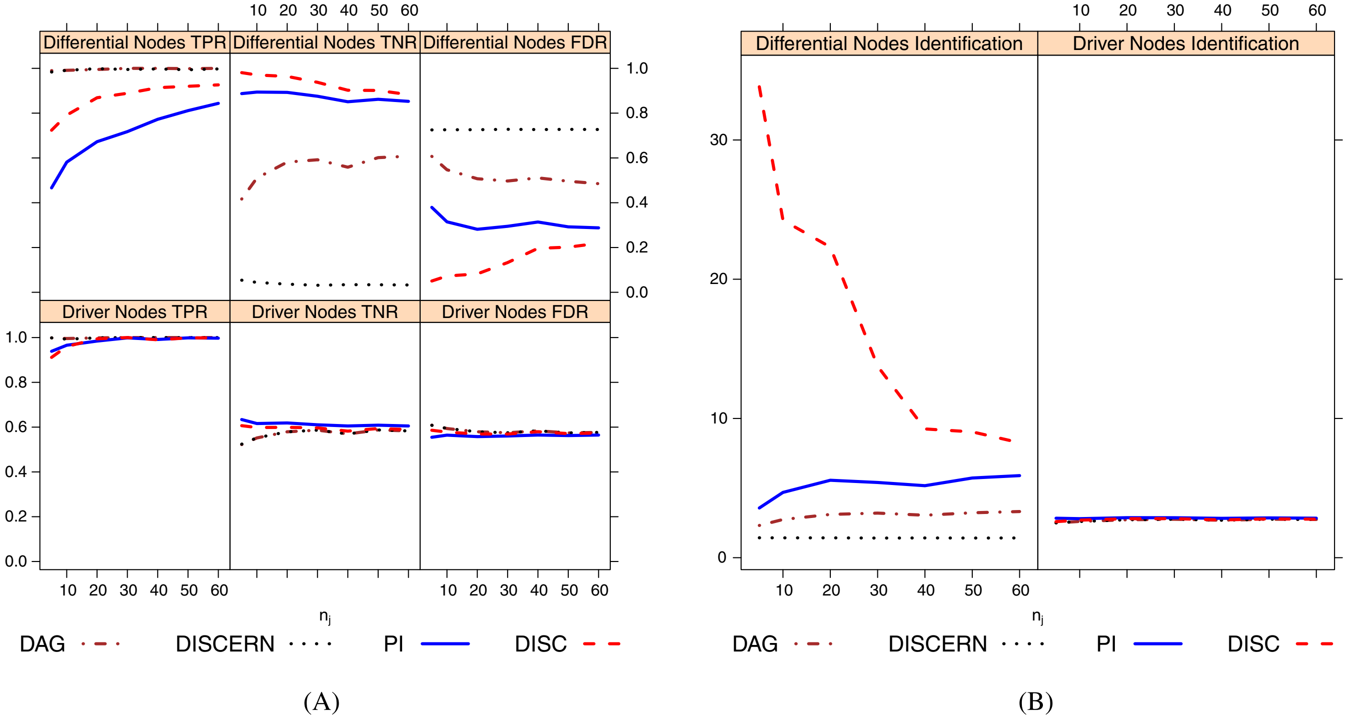


**FIGURE C1** Panel (a) shows averages of TPR, TNR, and FDR across 100 Monte Carlo replicates in terms of differential nodes identification (upper half) and driver nodes identification (lower half) associated with four methods, DAG (dash-dotted lines), DISCERN (dotted lines), PI (sold lines), and DISC (dashed lines). Panel (b) shows the corresponding averages of the ratio, (TPR+TNR)/FDR, for differential nodes identification (left half) and driver nodes identification (right half).