Biometrics 000, 000–000 DOI: 000
000 0000

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

Yan Zhou*

Gilead Sciences, Foster City, California

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

and

Peter X.-K. Song*

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

**email:* pxsong@umich.edu

and

Xiaoquan Wen*

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

**email:* xwen@umich.edu

Summary: This document provides supplementary materials concerning further details required for the understanding of the proposed methodology in paper "Structural factor equation models for causal network construction via directed acyclic mixed graphs".

Key words: Directed acyclic graph; directed acyclic mixed graphs; structural factor equation models; latent factors

This paper has been submitted for consideration for publication in *Biometrics*

1. Coordinate Descent Algorithm

We implement an efficient algorithm to yield the optimal solution that minimizes the L_1-norm penalized loss function in equation (3) of the main text under a fixed positive definite matrix $W = BB^T + \Psi$. Since minimizing this penalized loss function with respect to Θ is equivalent to a convex optimization problem, the objective function decreases over iterations, and the algorithmic convergence is warranted (Tseng and Yun, 2009). We first reformulate the optimization, so that the penalized loss function reduces to a regular LASSO regression problem with $\xi_{ij} = 1$ and $c_{ij} = 1$ in the penalized loss function. Then, we apply the following active-shooting algorithm to find the sparse solution of Θ efficiently.

Given $Y_{P\times N}^T = (\mathbf{y}_1^T, \ldots, \mathbf{y}_N^T)$ and $\widetilde{Y} \triangleq YW^{-1/2}$ with $W = BB^T + \Psi$, it is easy to see that the quadratic loss function $\frac{1}{2N}\sum_{n=1}^{N}(\mathbf{y}_n - \Theta\mathbf{y}_n)^T(BB^T + \Psi)^{-1}(\mathbf{y}_n - \Theta\mathbf{y}_n)$ equals to $\frac{1}{2N} \parallel \mathcal{Y} - \mathcal{X}\boldsymbol{\beta} \parallel^2$, where $\boldsymbol{\beta} = (\theta_{21}, \ldots, \theta_{P1}, \ldots, \theta_{PP-1})^T$, $\mathcal{Y} = (\widetilde{Y}_2^T, \ldots, \widetilde{Y}_P^T)^T$, and $\mathcal{X} = (\mathcal{X}_{(2,1)}, \ldots, \mathcal{X}_{(P,P-1)})$ is an $N(P-1)$ by $\frac{P(P-1)}{2}$ matrix with

$$\mathcal{X}_{(i,\ j)} = (\underset{1^{\text{st}}\text{ block}}{0}, \ldots, \underset{(i-1)^{\text{th}}\text{ block}}{\widetilde{Y}_j^T}, \ldots, \underset{(P-1)^{\text{th}}\text{ block}}{0})^T.$$

Thus, the L_1-norm minimization in equation (3) of the main text is equivalent to the following optimization:

$$\min_{\Theta} \frac{1}{2N} \parallel \mathcal{Y} - \mathcal{X}\boldsymbol{\beta} \parallel^2 + \lambda \sum_{h=1}^{\frac{P(P-1)}{2}} \xi_h |c_h \beta_h|, \tag{1}$$

with ξ_h being the h-th element of vector $\boldsymbol{\xi} = (\xi_{21}, \ldots, \xi_{PP-1})^T$ and c_h being the h-th element of vector $\mathbf{c} = (c_{21}, \ldots, c_{PP-1})^T$. The dimensions of $\mathcal{Y}$ and $\boldsymbol{\beta}$ are $N(P-1)$ and $P(P-1)/2$, respectively, which are larger than N and P. This could involve significant computational burden. Note that $\mathcal{X}$ is a block matrix with many zero blocks, and utilizing its structural features in computation can help run the LASSO optimization algorithm more efficiently. Thus, to improve the computational efficiency, we implement the active-shooting method (Friedman et al., 2007, 2010; Peng et al., 2009) in the coordinate descent algorithm. It can be shown that the resulting computational complexity of solving (1) reduces to as low as

the order of $\min(O(NP^2), O(P^3))$, which is equivalent to performing P individual LASSO regressions in the neighborhood selection method (Shojaie and Michailidis, 2010).

Unlike the neighborhood selection method (Shojaie and Michailidis, 2010) which imposes sparsity on individual neighborhoods during optimization, in our method the sparsity of $\boldsymbol{\beta}$ is treated in a global fashion via the regularized objective function (1). Thus, our approach utilizes the data more efficiently, and seems more natural to deal with networks with clustered hubs that are typically formed around master regulators. Indeed, detecting master regulators and their surrounding network structures is of great interest in the reconstruction of gene regulatory networks. In addition, given certain established knowledge of directed edges, the proposed regularization method in (1) has the flexibility of incorporating such information. That is, we can determine whether or not to penalize a pair of nodes by including suitable entries in the weighting term $\mathbf{c}$. Also with the utilization of the term $\boldsymbol{\xi}$, we can assign different adaptive weights to different pairs of nodes according to their importance.

The active-shooting algorithm proceeds as follows: at each updating step, we first define an "active" set of currently nonzero coefficients and update the coefficients within the active set until convergence is achieved before moving on to update other parameters. This is computationally appealing because the active set usually remains small under the sparse model assumption. Defining a current active set $H = \{h : c_h\beta_h \neq 0\}$, we update $\beta_{h_0 \in H}$ by the following operation in (2) with all other $\beta_{h \neq h_0}$ fixed until convergence is achieved in H.

$$\hat{\beta}_{h_0} = \begin{cases} (\mathcal{Y} - \sum_{h \neq h_0} \beta_h \mathcal{X}_h)^T \mathcal{X}_{h_0} / \|\mathcal{X}_{h_0}\|_2^2, \text{ if } c_{h_0} = 0, \\ S\Big((\mathcal{Y} - \sum_{h \neq h_0} \beta_h \mathcal{X}_h)^T \mathcal{X}_{h_0},\ N\lambda\xi_{h_0}\Big) / \|\mathcal{X}_{h_0}\|_2^2, \text{ if } c_{h_0} = 1, \end{cases} \tag{2}$$

where $S(a, b) = sgn(a)(|a| - b)_+$ is the soft-thresholding operator.

2. Some additional simulation results

Table 1 lists some additional results of both simulation experiments I and II in the paper, where we further compare the proposed SFEM with PC-algorithm (given bonferroni corrected α) and the score-based method (sparsebn). Table 1 suggests that the proposed SFEM method with the number of latent factors K selected via eigenvalue ratio criterion shows a satisfactory performance with the highest sensitivity and MCC as well as the lowest false discovery ($FP + FN$).

[Table 1 about here.]

We further extend the above 50 simulations for Simulation II (i.e. $P = 200, M = 100, N = 100, K_{\text{true}} = 5, PEV = 1 : 4$) to 500 simulations. The following plot is the spaghetti plot for each method. With no doubt, SFEM with full knowledge on node order outperforms all the other methods, especially the PC algorithm or the score-based method (sparsebn), which do not utilize node order information and also do not adjust for latent factors. When comparing SFEM with partial knowledge versus with no knowledge, we find that there is an improvement on network identification because of utilizing the existing partial knowledge of node order. When comparing the variation of SFEM across full knowledge, partial knowledge and no knowledge about node order respectively, we can see that SFEM with full knowledge has the smallest variation. This is probably due to the model identifiability. Once we do not know the node order (i.e. partial knowledge and no knowledge), this identifiability issue is reflected via larger variations.

[Figure 1 about here.]

3. Some Empirical Results on Computational Complexity

Fitting the SFEM involves two separate operations. One is related to a factor analysis of residuals, which is implemented by the EM algorithm; and the other is the operation of

coordinate descent algorithm to search for sparse adaptive lasso solution in the estimation of the weighted adjacency matrix Θ. In our computation, given the number of latent factors K and the tuning parameter λ, when the weighted adjacency matrix Θ is fixed, the computational complexity is $O(NPK)$ per iteration in the estimation of the factor loadings B and uniqueness $\Psi = \sigma^2 I_P$. When the covariance matrix $W = B^T B + \Psi$ is given, the computational cost of solving Θ by the sparse adaptive lasso via the popular active shooting method in the coordinate descent algorithm is $min(O(NP^2), O(P^3))$. Here N is the sample size, P is the number of nodes, and K is the number of latent factors. To demonstrate the actual run-time in model fitting, here we present a simulation experiment focusing on computation time. Using the Simulation II setup outlined in the paper ($P = 200$, $K = 5$, $N = 100$, $M = 100$, $PEV = 1 : 4$), we report the computation time in various scenarios in Table 2 in terms of average running time in seconds over 50 simulations to solve SFEM under the selected tuning parameter λ. All calculations were carried out on a computer with an Intel Xeon 2.30 GHz processor.

[Table 2 about here.]

With no surprise the computational cost increases along the increase in the number of latent factors K. This is because the more complicated the factor model is the heavier computational burden the EM algorithm encounters to estimate loading coefficients. In practice, instead of trying a wide range of K values, one may narrow down such range by identifying the top K eigenvalues of sample covariance matrix of Y or apply eigenvalue ratio (ER) criterion to select the number of latent factors K. At this moment this strategy is learned from our empirical experience, which needs further theoretical investigation.

4. Analysis of cell signaling data

This section demonstrates an application of the proposed SFEM method to analyze multivariate flow cytometry data available in Sachs et al. (2005), which has been previously analyzed by Shojaie and Michailidis (2010); Fu and Zhou (2013); Friedman et al. (2008); Aragam and Zhou (2015), among others. This dataset includes 11 phosphorylated proteins from $N = 7466$ cells. The consensus network, constructed by experimental annotations, has 20 edges, which is displayed in Figure 2 and is used as the benchmark to assess the accuracy of an estimated network structure. A direction from node i to node j is interpreted as a causal influence from protein i to protein j. Following Shojaie and Michailidis (2010), the node ordering in the DAG is treated as *a priori* feature among 11 proteins.

[Figure 2 about here.]

Based on the scree plot (not shown) and the eigenvalue ratio (ER) method, we obtain $K_{ER} = 4$. The optimal tuning parameter is determined by the 5-fold cross-validation method.

We explore the SFEM under different numbers of latent factors $K = 0, 2, 4, 6$, where $K = 0$ corresponds to the analysis given by Shojaie and Michailidis (2010) and $K = 4$ is the estimated number of latent factors. Figure 3 shows the plot of the number of correctly detected edges versus the total number of detected edges across different number of latent factors. Comparing the results obtained under $K = 0, 2, 4, 6$, we find that the SFEM with the estimated $K_{\mathrm{ER}} = 4$ performs slightly better than the other cases. To compare the SFEM $(K = 0)$ with the SFEM$(K = 4)$ in the case where both methods detect 25 edges, 10 edges detected by the latter are in the consensus network as opposed to 7 edges detected by the former in the consensus network. So, adjusting the latent factors can improve the sensitivity by $3/25 = 12\%$.

[Figure 3 about here.]

We display the estimated DAGs in Figure 4. Major differences between these two DAGs lie

in the domain of false discoveries. For example, among the total of 25 directed edges, the SFEM with $K_{ER} = 4$ reports 15 false positives in comparison to 18 false positives from the SFEM with $K = 0$. Hence, the SFEM with $K_{ER} = 4$ gives a more reliable analysis, with fewer false positives and more true positive signals in comparison to the SFEM with $K = 0$.

[Figure 4 about here.]

Nevertheless, several known edges are not detected by both SFEMs with $K = 0$ and $K_{ER} = 4$. One possible reason is that the proposed SFEM is developed for a linear Bayesian network, which may not be able to detect nonlinear causal relationships.

[Table 3 about here.]

In addition, we also compare the proposed SFEM with other popular methods, such as the PC-algorithm and the score-based method, when 25 edges are detected. To make a fair comparison, we further simply ignore the edge direction in the comparison for the PC algorithm and the score-based method. In other words, no matter $i \to j$ or $j \to i$ is detected by the PC-algorithm or the score-based method, we will count it as a positive finding.

[Table 4 about here.]

Obviously, the SFEM with $K_{ER} = 4$ performs equivalently well as PC-algorithm and the score-based method under this large sample scenario (i.e., $N = 7466 >> P = 11$) and outperforms other methods in terms of higher sensitivity and MCC.

5. Additional analysis results of METABRIC gene expression data

A. We first analyze the reordered METABRIC gene expression data via our proposed SFEM method where K ranges from 0 to 5. Figure 5 shows that with an increase in the number of latent factors, the detected number of edges decreases.

[Figure 5 about here.]

When fully ignoring latent factors ($K = 0$), a total of 211 edges are detected via our method. However, most of these detected edges are deemed false positive and are not biologically meaningful. See Figure 6 (a). In addition, the eigenvalue ratio method suggests that the number of latent factors is $K = 2$ and we detect a total of 170 edges accordingly. See Figure 6 (b). In addition to the SFEM method, we analyze this METABRIC gene expression data via another two popular methods: PC-algorithm and score-based method. For the PC-algorithm (given Bonferroni corrected $\alpha = 7.53e - 06$), we totally detect 133 edges. See Figure 6 (c). For the score-based method, we totally detect 215 edges, as shown in Figure 6 (d). To sum up, Figure 7 shows 38 common edges among 42 genes collectively detected by the different methods.

[Figure 6 about here.]

[Figure 7 about here.]

B. We apply our SFEM method on 50 bootstrap samples, where the final gene regulatory network is drawn by the majority voting strategy; that is, a final edge is reported when it is detected at least 50% chance out of 50 bootstrap samples. The frequency of the number of latent factors K determined by the eigenvalue ratio method among 50 bootstrap samples is summarized in Table 5.

[Table 5 about here.]

In the finally voted network, we detect 125 causal relationships among 71 genes, which is illustrated in Figure 8.

[Figure 8 about here.]

In addition, we apply the PC-algorithm and the sparsebn method on 50 bootstrap samples. In the final causal network, in which a detection is called if it occurs in more than 25 bootstrap samples, the PC-algorithm has detected 111 causal relationships among 74 genes, whereas

the score-based method has detected 153 causal relationships among 79 genes. The detailed final GRNs are illustrated in Figure 9.

[Figure 9 about here.]

To summary, we detect 60 common edges among 56 genes, as shown in Figure 10 which represents a shared finding across the SFEM, the score-based method and the PC algorithm.

[Figure 10 about here.]

6. Computing Code

Our computing code used in the simulation studies is available online under the package name "SFEM" at webpage: `http://www.umich.edu/~songlab/software.html`. In this sample coding package, we have provide the following items to test our computing code:

1. `Y_1.txt` is the example dataset. This dataset is generated under the settings of Simulation I, where $N = 25, P = 50$ with the total true edges $M = 25, K = 2, PEV = 1 : 3$.
2. `Theta_simple.txt` is the true weighted adjacency matrix for Simulation I. This is a lower triangular matrix with total M=25 nonzero elements.
3. `main_K_2.R` is the main calculation function. It will calls other functions: EM.R, OLS.R, Theta_initialization.R, and Parameter_estimation.R. This runs the SFEM method on the dataset `Y_1.txt` under $K = 2$ for a given tuning parameter $\gamma = 2$. We also compare the estimated network adjacency matrix with the true network adjacency matrix, where the result is summarized as follows:

```
total_detect TP FP FN Sen     Spec       MCC          FPR
          26 25  1  0   1 0.999596 0.9803826 0.0004040404
```

References

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

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics* **1,** 302–332.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9,** 432–441.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33,** 1–22.

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

Peng, J., Wang, P., Zhou, N., and Zhu, J. (2009). Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association* **104,** 735–746.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science* **308,** 523–529.

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

Tseng, P. and Yun, S. (2009). A block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. *Journal of Optimization Theory and Applications* **140,** 513–535.

Figure 1: Spaghetti plot based on Simulation II with $P = 200, M = 100, K = 5, N = 100, PEV = 1 : 4$ as well as the estimated $K_{ER} = 5$ (100%). The x-axis is the total number of detected edges, and the y-axis is the number of correctly identified edges averaged over 500 replicates.

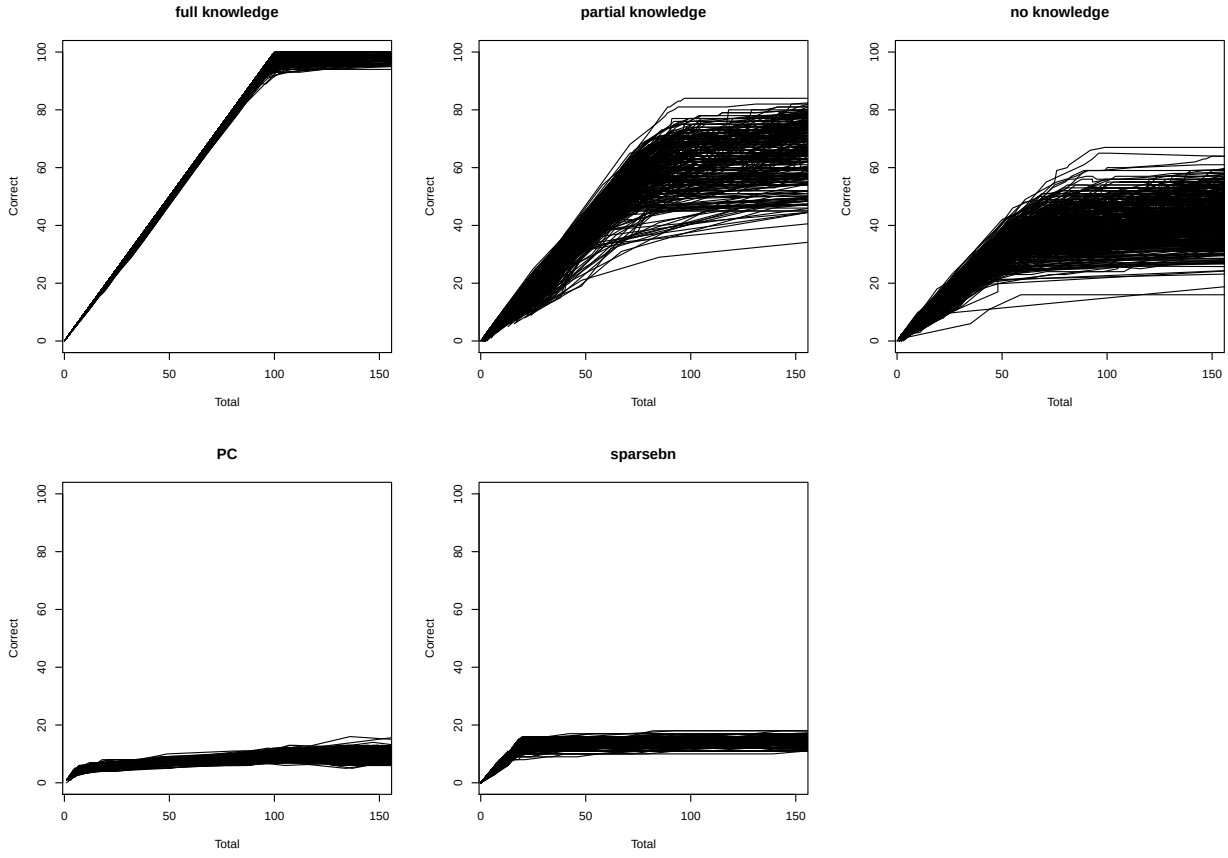

Figure 2: A benchmark DAG based on the consensus signaling network of 11 proteins.

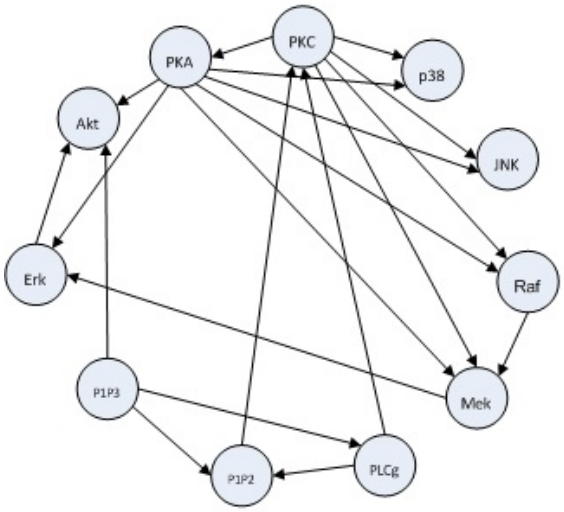

Figure 3: Results from the analysis of cell signaling data by the proposed SFEM with $K = 0, 2, 4, 6$ with $K_{ER} = 4$. The x-axis is the total number of detected edges, and the y-axis is the number of correctly identified edges.

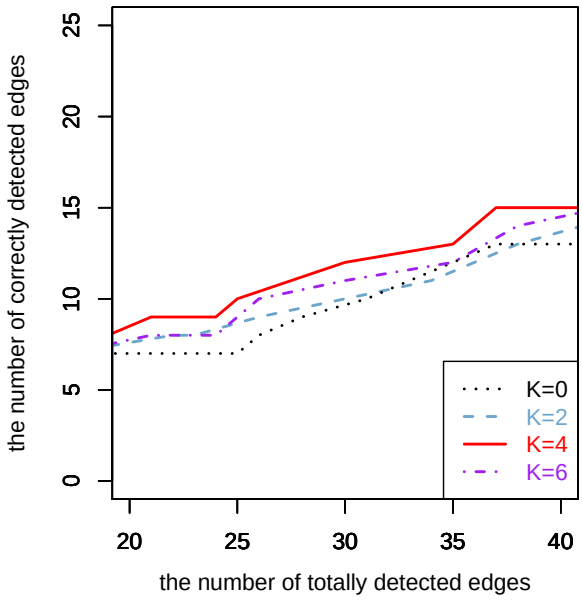

Figure 4: Estimated causal interactions among 11 proteins of the signaling pathway. Black squares represent TP, pink squares represent FN, and grey squares represent FP. The right panel (b) also displays the loading matrix of 4 common latent factors $z_1, \ldots, z_4$, where white represents a loading coefficient smaller than 0.1.

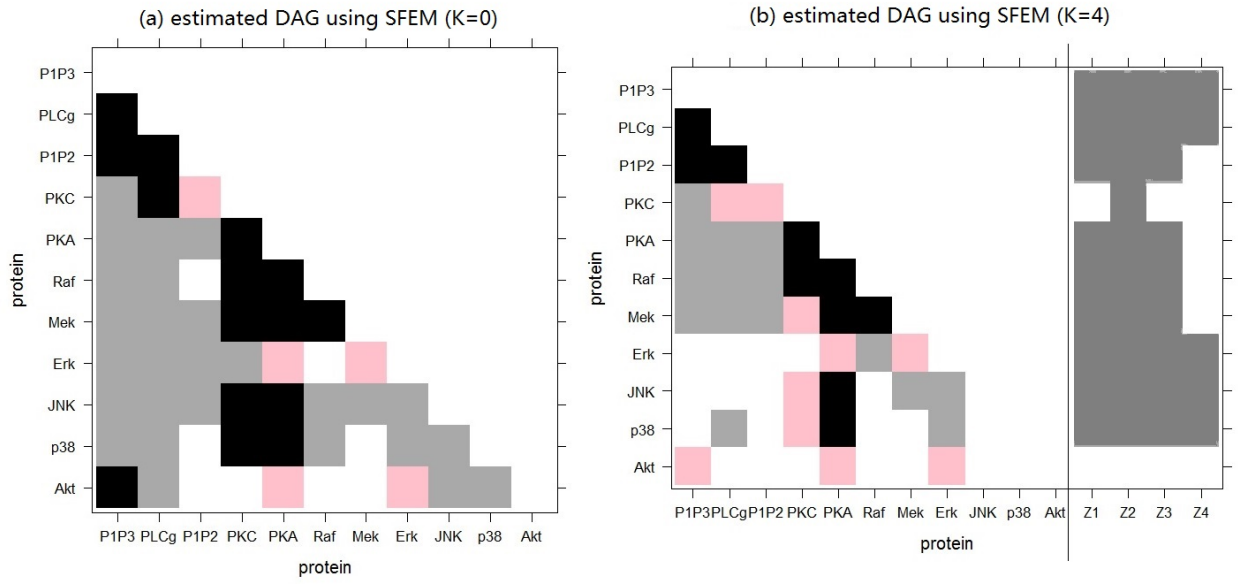

Figure 5: The number of detected edges under different K.

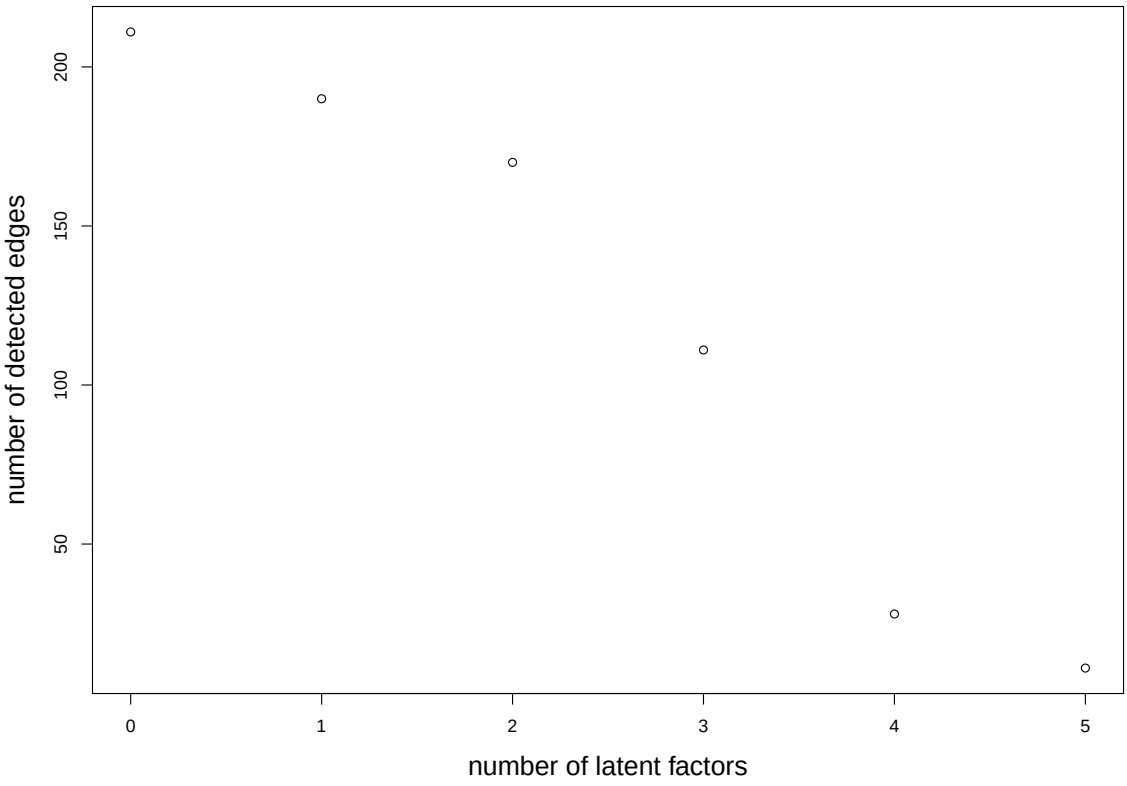

Figure 6: Estimated gene regulatory network using different methods.

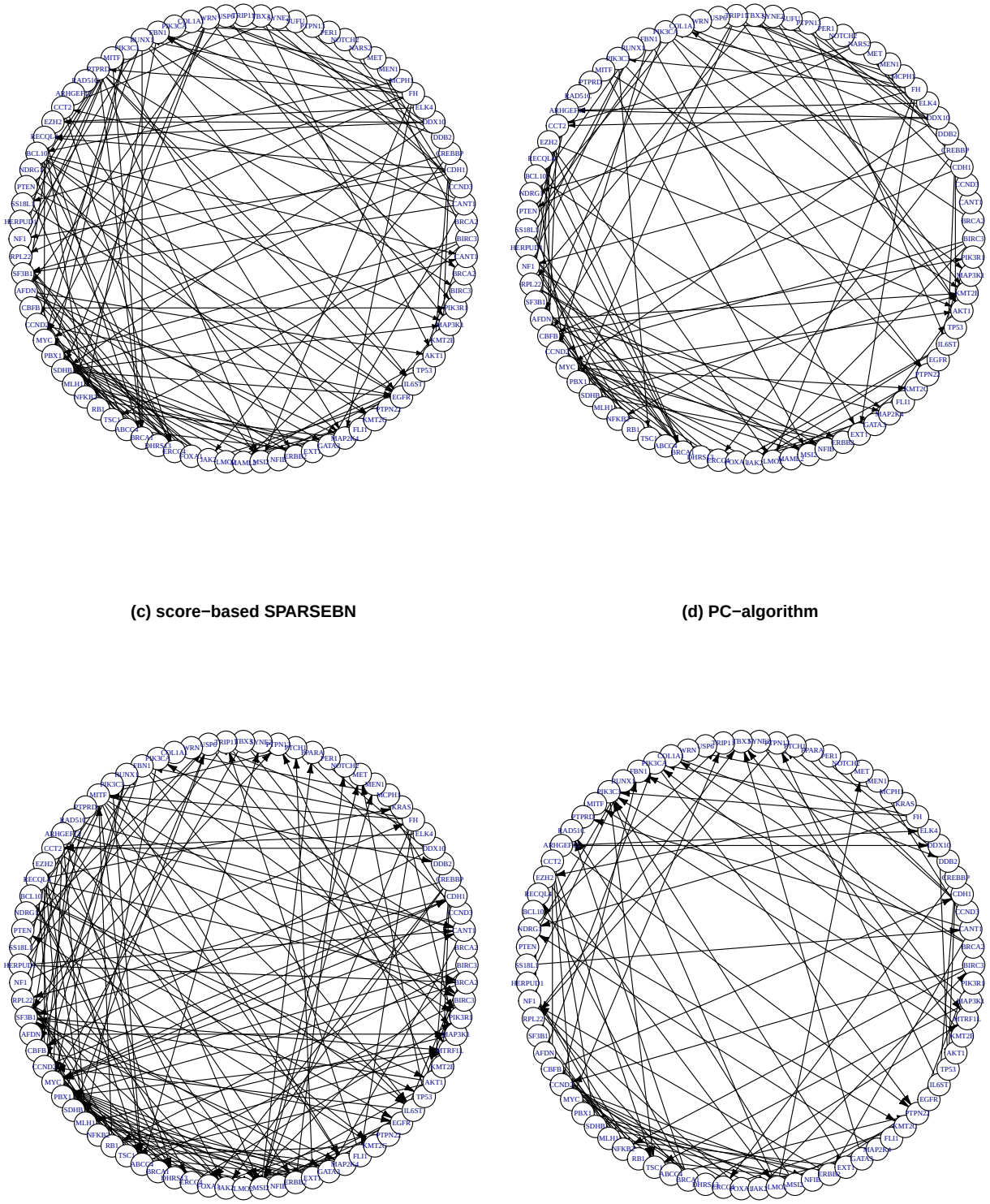

Figure 7: 38 causal relationships among 42 genes across different methods: $SFEM_{K=0}$, $SFEM_{K=2}$, PC algorithm and SPARSEBN.

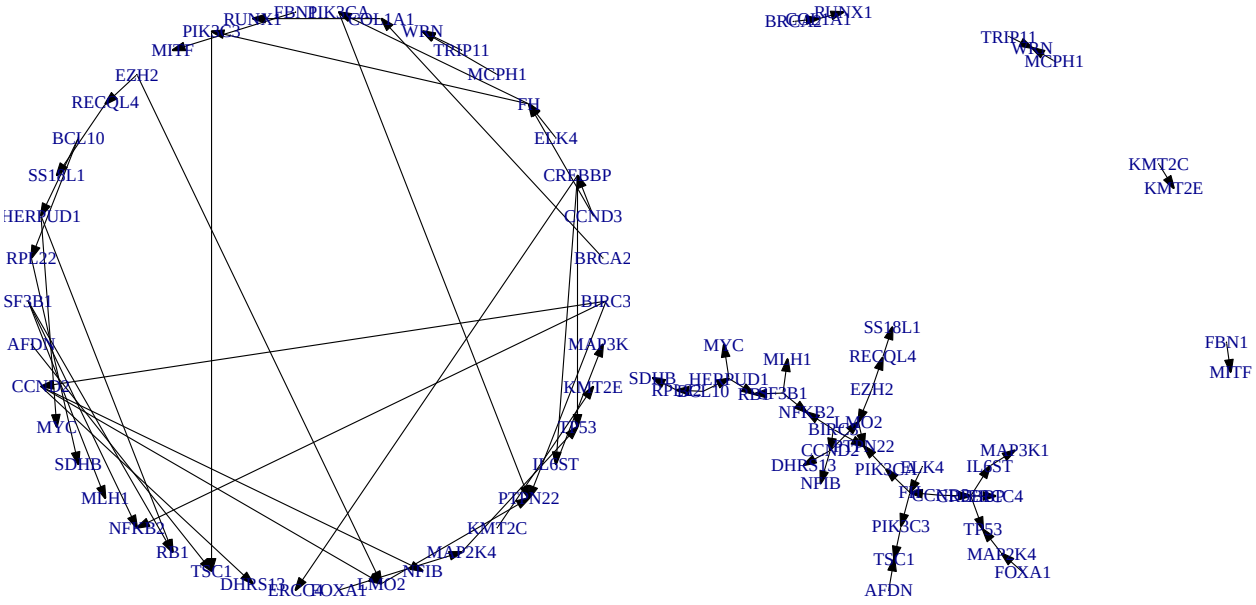

Figure 8: Estimated causal interactions among 71 driver genes of breast cancer.

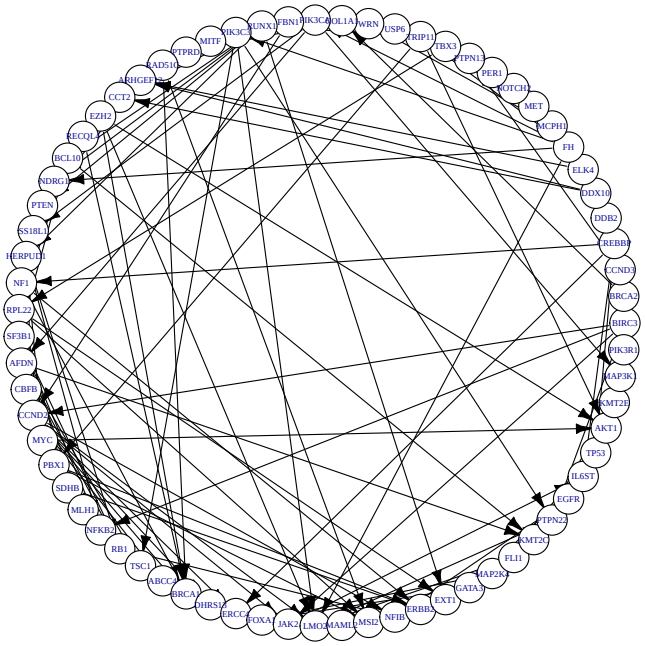

Figure 9: Final estimated gene regulatory network among 50 bootstrap samples via PC algorithm and SPARSEBN method

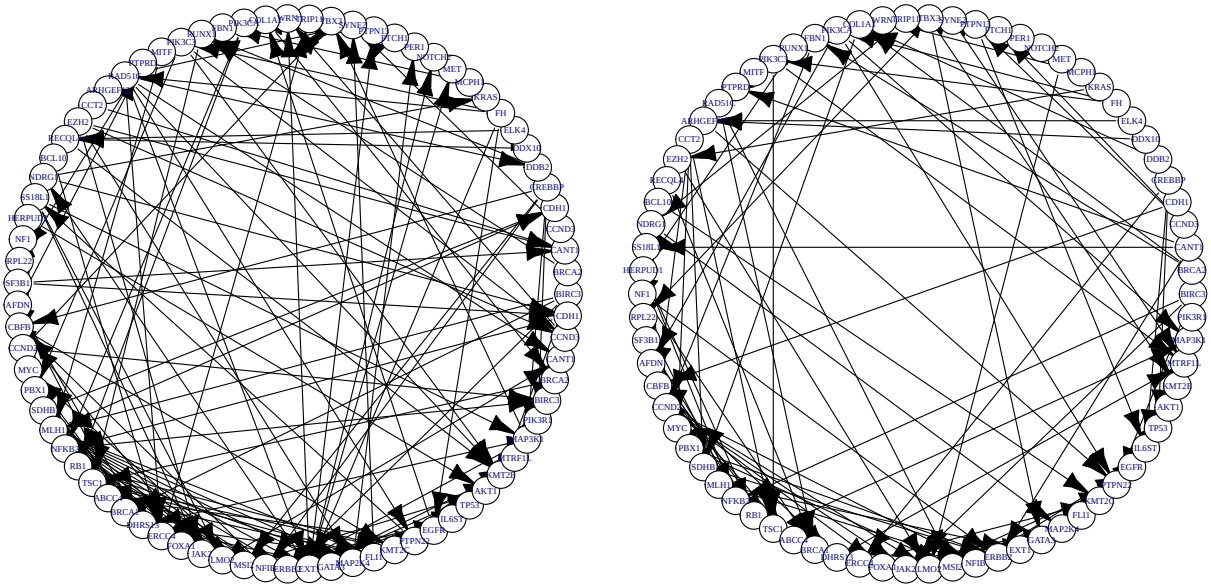

Figure 10: 60 causal relationships among 56 genes over 50 bootstrap samples across different methods: SFEM, PC algorithm and SPARSEBN.

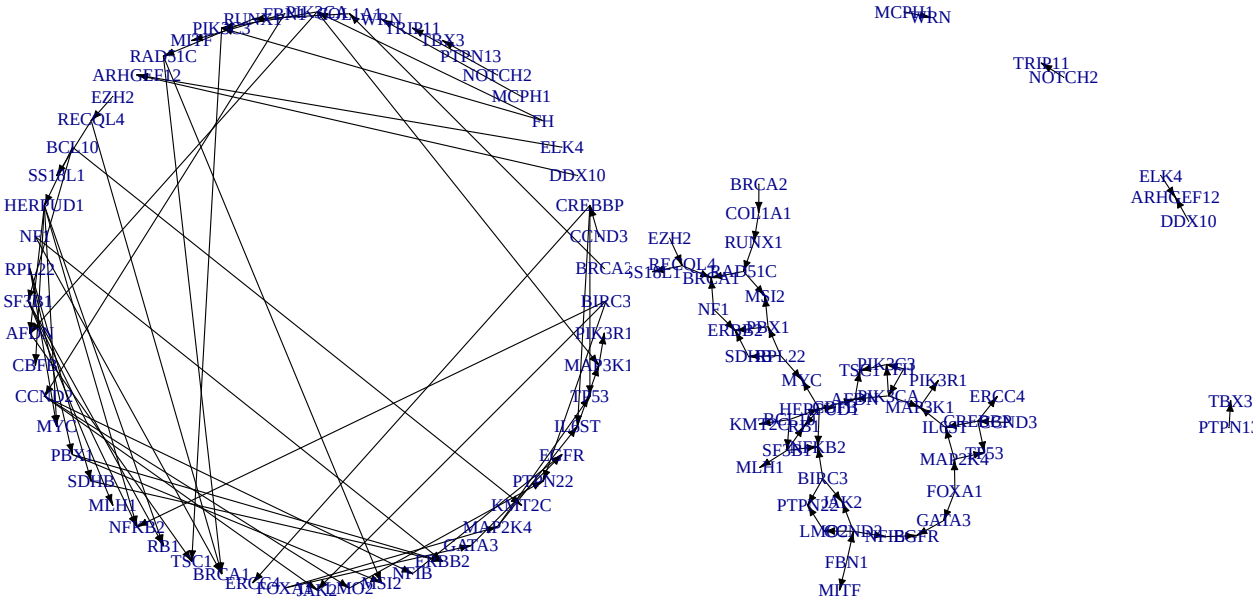

Table 1: Additional results to compare SFEM with PC-algorithm and score-based method (sparsebn) over 50 simulations: small and large DAG simulation designs, respectively.

PEV	K_{true}	Method	Total (TP+FP)	TP	FP	FN	Sen	MCC	K_{ER}(%)
Simulation I									
1:3	2	SFEM_{ER}	29.44	24.76	4.68	0.24	0.99	0.92	2 (100%)
		PC-algorithm	15.64	10.68	4.96	14.32	0.43	0.53	
		SPARSEBN	260.54	9.88	250.66	15.12	0.40	0.10	
Simulation II									
1:4	5	SFEM_{ER}	104.04	98.12	5.92	1.88	0.98	0.96	5 (100%)
		PC-algorithm	109.36	9.40	99.96	90.60	0.09	0.09	
		SPARSEBN	347.30	11.60	335.70	88.40	0.12	0.06	

Table 2: An average running time over 50 simulations in seconds to solve SFEM under the selected tuning parameters.

Method	K=0	K=5	K=10
SFEM	179.93 (44.45)	440.65 (91.77)	1186.69 (290.74)

Table 3: Comparison between SFEM with $K = 0$ and $K_{\mathrm{ER}} = 4$ under the selected optimal tuning parameter.

Method	Total	TP	FP	FN	Method	Total	TP	FP	FN
K=0	42	15	27	5	K=4	25	10	15	10

Table 4: Comparison among the PC algorithm, the score-based method, SFEM with $K = 0$ and $K = 4$ when 25 edges are detected.

Method	Total (TP+FP)	TP	FP	FN	Sen	MCC
PC-algorithm with direction	25	10	15	10	0.50	0.32
PC-algorithm ignoring direction	24	10	14	10	0.50	0.34
SPARSEBN with direction	25	5	20	15	0.25	0.05
SPARSEBN ignoring direction	25	9	16	11	0.45	0.27
$\text{SFEM}_{K=0}$	25	7	18	13	0.35	0.16
$\text{SFEM}_{K=4}$	25	10	15	10	0.5	0.32

Table 5: The frequency of K determined by the eigenvalue ratio method among 50 bootstrap samples.

	K=1	K=2	K=3	K=4	K=5
Count (%)	5(10%)	25 (50%)	0 (0%)	17 (34%)	3 (6%)