Supplementary Materials
SpaceLog: an R package for inferring gene-gene networks using SPACE model with log penalty

Qian (Vicky) Wu, Wei Sun, and Li Hsu

S1 Methods

S1.1 Active-shooting

We adapted the same idea `active-shooting` algorithm from Peng *et al.* (2009) to update the coefficient estimation iteratively in `space-log`. Without loss of generality, we kept most notation from Peng *et al.* (2009) but tailored with `space-log`. Let $\mathcal{Y} = (y_1, y_2, ..., y_n)$, $X = (x_{ij})_{n\times m} = (Y_1, Y_2, ..., Y_m)$, and the penalized loss function is $f(\mathcal{Y}, \beta) = \frac{1}{2}\Sigma_{i=1}^{m} w_i||\mathcal{Y} - X\beta||^2 + \Sigma_{i\neq j} p(|\rho_{i,j}|; \lambda, \tau)$. Since $\beta_{ij} = \rho_{i,j}\sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}}$, minimizing the penalized loss function to find $\beta = \text{argmin}_\beta f(\beta)$ is equivalent to find $\rho = \text{argmin}_\rho f(\rho)$ for a given λ, τ.

We standardized each gene such that $mean(Y_i) = 0$ and $var(Y_i) = 1$, for $i = 1, \ldots, m$. Denote $\xi_i = Y_i^T Y_i$. We have

$$\begin{aligned} X_{(i,j)}^T X_{(i,j)} &= \xi_j \frac{\sigma^{jj}}{\sigma^{ii}} + \xi_i \frac{\sigma^{ii}}{\sigma^{jj}}; \\ \mathcal{Y}^T X_{(i,j)} &= \sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}} Y_i^T Y_j + \sqrt{\frac{\sigma^{ii}}{\sigma^{jj}}} Y_j^T Y_i. \end{aligned} \tag{1}$$

1. Initialization
Let

$$\begin{aligned} \rho_{(i,j)}^{(0)} &= \frac{\left(|\mathcal{Y}^T X_{(i,j)}| - \frac{\lambda}{|\rho_{(i,j)}^{(old)}|+\tau}\right)_+ \cdot \text{sign}(\mathcal{Y}^T X_{(i,j)})}{X_{(i,j)}^T X_{(i,j)}} \\ &= \frac{\left(\left|\sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}} Y_i^T Y_j + \sqrt{\frac{\sigma^{ii}}{\sigma^{jj}}} Y_j^T Y_i\right| - \frac{\lambda}{|\rho_{(i,j)}^{(old)}|+\tau}\right)_+ \cdot \text{sign}(Y_i^T Y_j)}{\xi_j \frac{\sigma^{jj}}{\sigma^{ii}} + \xi_i \frac{\sigma^{ii}}{\sigma^{jj}}} \end{aligned} \tag{2}$$

where $|\rho_{(i,j)}^{(old)}|$ was set to be zero for initial step and $\frac{\lambda}{|\rho_{(i,j)}^{(old)}|+\tau} = \frac{\lambda}{\tau}$.

$$\widehat{Y}_j^{(0)} = \left(\sqrt{\frac{\sigma^{11}}{\sigma^{jj}}}Y_1, ..., \sqrt{\frac{\sigma^{pp}}{\sigma^{jj}}}Y_p\right) \cdot \begin{pmatrix} \rho_{(1,j)}^{(0)} \\ \vdots \\ \rho_{(p,j)}^{(0)} \end{pmatrix} \tag{3}$$
$$E^{(0)} = \mathcal{Y} - \widehat{\mathcal{Y}}^{(0)} = \left((E_1^{(0)})^T, ..., (E_p^{(0)})^T\right)$$

where $E_j^{(0)} = Y_j - \widehat{Y}_j^{(0)}$.

2. Update $\rho_{(i,j)}^{(0)} \longrightarrow \rho_{(ij)}^{(1)}$

Let

$$\begin{aligned} A_{(i,j)} = &\quad (E_j^{(0)})^T \cdot \sqrt{\frac{\sigma^{ii}}{\sigma^{jj}}}Y_i \\ A_{(j,i)} = &\quad (E_i^{(0)})^T \cdot \sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}}Y_j \end{aligned} \tag{4}$$

Then

$$\begin{aligned} (E^{(0)})^T X_{(i,j)} &= (E_i^{(0)})^T \cdot \sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}}Y_j + (E_j^{(0)})^T \cdot \sqrt{\frac{\sigma^{ii}}{\sigma^{jj}}}Y_i \\ &= A_{(j,i)} + A_{(i,j)} \end{aligned} \tag{5}$$

And

$$\begin{aligned} \rho_{(i,j)}^{(1)} &= \text{sign}\left(\frac{(E^{(0)})^T X_{(i,j)}}{X_{(i,j)}^T X_{(i,j)}} + \rho_{(i,j)}^{(0)}\right)\left(\left|\frac{(E^{(0)})^T X_{(i,j)}}{X_{(i,j)}^T X_{(i,j)}} + \rho_{(i,j)}^{(0)}\right| - \frac{\lambda/(|\rho_{(i,j)}^{(0)}| + \tau)}{X_{(i,j)}^T X_{(i,j)}}\right)_+ \\ &= \text{sign}\left(\frac{A_{(j,i)} + A_{(i,j)}}{\xi_j \frac{\sigma^{jj}}{\sigma^{ii}} + \xi_i \frac{\sigma^{ii}}{\sigma^{jj}}} + \rho_{(i,j)}^{(0)}\right)\left(\left|\frac{A_{(j,i)} + A_{(i,j)}}{\xi_j \frac{\sigma^{jj}}{\sigma^{ii}} + \xi_i \frac{\sigma^{ii}}{\sigma^{jj}}} + \rho_{(i,j)}^{(0)}\right| - \frac{\lambda/(|\rho_{(i,j)}^{(0)}| + \tau)}{\xi_j \frac{\sigma^{jj}}{\sigma^{ii}} + \xi_i \frac{\sigma^{ii}}{\sigma^{jj}}}\right)_+ \end{aligned} \tag{6}$$

3. Update $\rho^{(k)} \longrightarrow \rho^{(k+1)}$

From the previous iteration, we have

- (i_0, j_0): index of coefficient that is updated in the previous iteration.
- $\rho_{(i,j)}^{(k)} = \begin{cases} \rho_{(i,j)}^{(k-1)} & \text{if } (i,j) \neq (i_0, j_0), \text{ nor } (j_0, i_0) \\ \rho_{(i,j)}^{(k-1)} - \Delta & \text{if } (i,j) = (i_0, j_0), \text{ or } (j_0, i_0) \end{cases}$

Then,

$$\begin{aligned} E_l^{(k)} &= E_l^{(k-1)} \text{ for } l \neq i_0, j_0; \\ E_{j_0}^{(k)} &= E_{j_0}^{(k-1)} + \widehat{Y}_{j_0}^{(k-1)} - \widehat{Y}_{j_0}^{(k)} \\ &= E_{j_0}^{(k-1)} + \textstyle\sum_{i=1}^p \sqrt{\frac{\sigma^{ii}}{\sigma^{j_0 j_0}}} Y_i (\rho_{(i,j_0)}^{(k-1)} - \rho_{(i,j_0)}^{(k)}) \\ &= E_{j_0}^{(k-1)} + \sqrt{\frac{\sigma^{i_0 i_0}}{\sigma^{j_0 j_0}}} Y_{i_0} \cdot \Delta; \\ E_{i_0}^{(k)} &= E_{i_0}^{(k-1)} + \sqrt{\frac{\sigma^{j_0 j_0}}{\sigma^{i_0 i_0}}} Y_{j_0} \cdot \Delta. \end{aligned} \tag{7}$$

Suppose the index of the coefficient we would like to update in this iteration is (i_1, j_1), then let

$$\begin{aligned} A_{(i_1,j_1)} &= (E_{j_1}^{(k)})^T \cdot \sqrt{\frac{\sigma^{i_1 i_1}}{\sigma^{j_1 j_1}}} Y_{i_1}, \\ A_{(j_1,i_1)} &= (E_{i_1}^{(k)})^T \cdot \sqrt{\frac{\sigma^{j_1 j_1}}{\sigma^{i_1 i_1}}} Y_{j_1}. \end{aligned} \tag{8}$$

We have

$$\begin{aligned} \rho_{(i,j)}^{(k+1)} = \quad & \text{sign}\left(\frac{A_{(j_1,i_1)}+A_{(i_1,j_1)}}{\xi_j \frac{\sigma^{j_1 j_1}}{\sigma^{i_1 i_1}} + \xi_{i_1} \frac{\sigma^{i_1 i_1}}{\sigma^{j_1 j_1}}} + \rho_{(i_1,j_1)}^{(k)} \right) \\ & \times \left(\left| \frac{A_{(j_1,i_1)}+A_{(i_1,j_1)}}{\xi_j \frac{\sigma^{j_1 j_1}}{\sigma^{i_1 i_1}} + \xi_{i_1} \frac{\sigma^{i_1 i_1}}{\sigma^{j_1 j_1}}} + \rho_{(i_1,j_1)}^{(k)} \right| - \frac{\lambda/(|\rho_{(i_1,j_1)}^{(k)}|+\tau)}{\xi_j \frac{\sigma^{jj}}{\sigma^{ii}} + \xi_i \frac{\sigma^{ii}}{\sigma^{jj}}} \right)_+ . \end{aligned} \tag{9}$$

S2 Tuning parameters

In penalized regression, we select the 'optimal' tuning parameter to get the minimal mean squared error between true parameter and penalized estimation. Cross-validation (CV) approach is the most commonly used method to choose the tuning parameter in penalized regression (Fan *et al.*, 2009); however, it is time-consuming and computationally expensive. Some recent literature (Wang *et al.*, 2009) has discussed that the BIC-type approach has better performance than cross-validation, so we didn't use CV for tuning parameter selection here. In this section, we compared several tuning parameter selection approaches, BIC (Zou *et al.*, 2007), extBIC (Chen and Chen, 2008, 2012), and oracle (Chen *et al.*, 2016) by extensive simulations. Note that for the NS approach, the tuning parameter is chosen for each gene and different genes may have different tuning parameter values but for the `space` approach, there is only one tuning parameter because of its joint modeling.

S2.1 BIC

BIC criterion is easy to compute for selecting the tuning parameters. For a given tuning parameter set θ, e.g., Lasso penalty (one tuning parameter $\theta = \lambda$) and log penalty (two tuning parameter $\theta = c(\lambda, \tau)$), the residual sum of squares for the ith regression is

$$RSS_i(\theta) = \Sigma_{k=1}^{n} (y_i^k - \hat{y}_i^k(\theta))^2 \tag{10}$$

And

$$BIC_i(\theta) = n \times \log(RSS_i(\theta)) + \log(n) \times nNon0_i \tag{11}$$

where $nNon0$ = number of non-zero coefficients for each regression. $BIC(\theta) = \Sigma_{i=1}^{m} BIC_i(\theta)$, and we select θ by minimizing $BIC(\theta)$

S2.2 extBIC

(Chen and Chen, 2008, 2012) proposed an extended BIC approach to select tuning parameters and showed that it had better performance than BIC (Chen *et al.*, 2016).

The extended BIC for the linear regression is

$$\begin{aligned} extBIC_i(\theta) = n \times \log(RSS_i(\theta)) + \log(n) \times nNon0_i + \\ 2 * (1 - \frac{1}{2\log(m)/\log(n)}) * \log(nComb) \end{aligned} \tag{12}$$

where $nComb$ = the number of combinations of nNon0 chosen from m-1 genes. $extBIC(\theta) = \Sigma_{i=1}^{m} extBIC_i(\theta)$, and we select θ by minimizing $extBIC(\theta)$

S2.3 Oracle

In order to compare the performance of BIC and extBIC, we included an 'oracle' approach to find the best tuning parameter. We used two metrics to show 'oracle': (a) Errors (FP+FN), which is the sum of number of false positive (FP) and false negative (FN), and the smaller the better; and (b) F1 score, which is the harmonic mean of precision and recall between 0 and 1, and the larger the better. We calculated the Error by

$$\begin{aligned} \text{Error}(\theta) &= \sum_{i=1}^{m} \text{Error}_i(\theta) \\ &= \sum_{i=1}^{m} nNon0_i + \widehat{nNon0}_i(\theta) - \cap(Non0_i, \widehat{Non0}_i(\theta)) \end{aligned} \tag{13}$$

where $Non0_i$ = genes with non-zero coefficient for the ith regression. We select θ by minimizing $Error(\theta)$ as the "oracle" tuning parameter for (a). We calculated the F1 score by

$$\begin{aligned} \text{F1}(\theta) &= \sum_{i=1}^{m} \text{F1}_i(\theta) \\ &= \sum_{i=1}^{m} \frac{2 \times precision_i(\theta) \times recall_i(\theta)}{precision_i(\theta) + recall_i(\theta)} \end{aligned} \tag{14}$$

We select θ by minimizing $F1(\theta)$ as the 'oracle' tuning parameter for (b). In the simulation, we used the same range of tuning parameter (λ and/or τ if applicable for θ) for each method and conducted a grid search to find the 'oracle' tuning parameters.

Figure S1 and S2 show the simulation results on the BA model (hub networks) with e=1 and the details can be found in S3 Simulation Studies. We compared BIC, extBIC, and Oracle across different GGN method (`NS-log`, `NS-lasso`, `space`, `space-log`) with sample size n=400 and different number

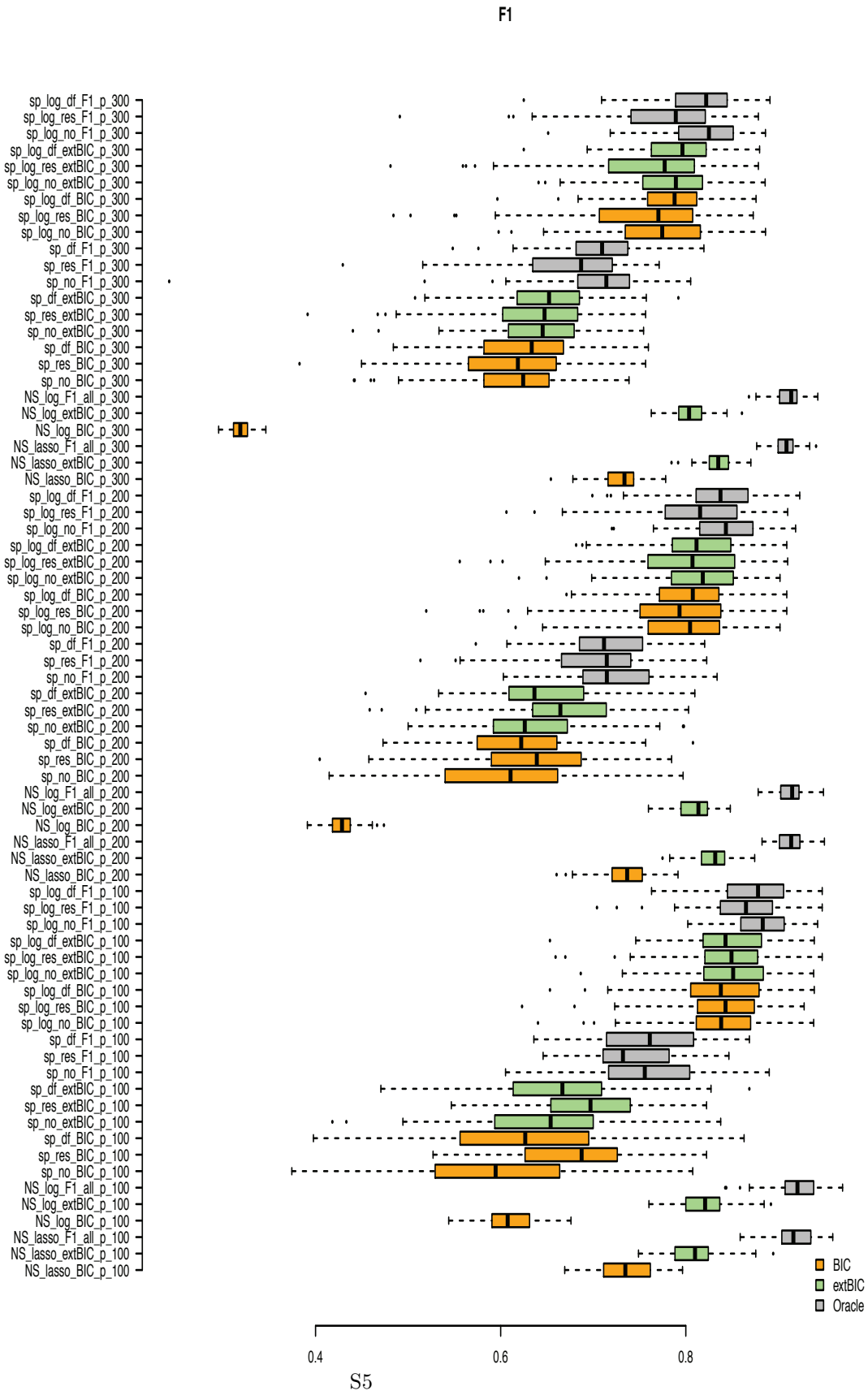
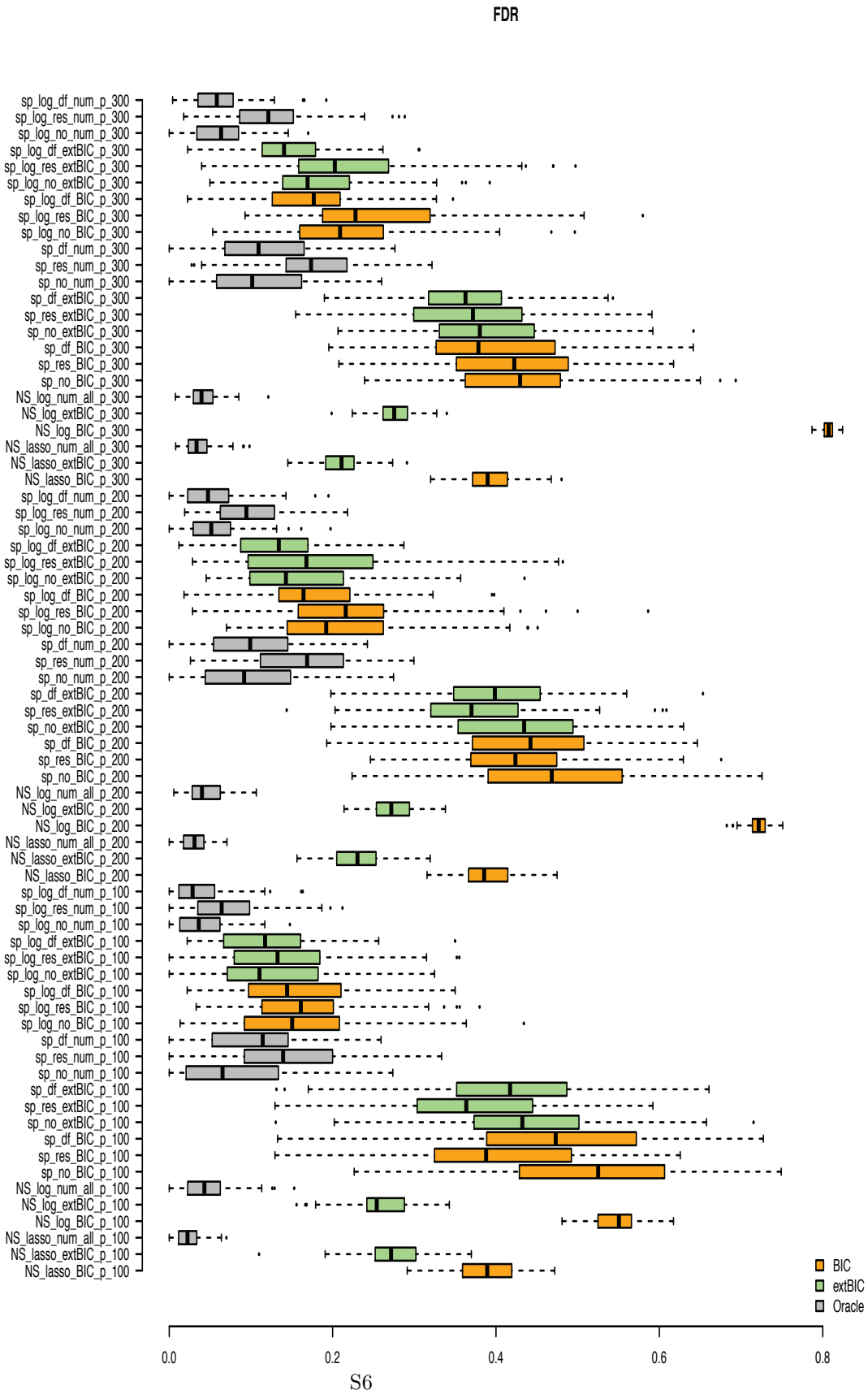


Figure S1: Compare BIC, extBIC, and Oracle by F1 score



S6

Figure S2: Compare BIC, extBIC, and Oracle by FDR rate

of genes m=100, 200, and 300. In Figure S1, we define the "oracle" criterion by maximizing F1 score. In Figure S2, we define the "oracle" criterion by minimizing Error defined in equation (13). In Figure S2, we illustrate the results of different approaches in terms of FDR. The results in terms of Error is similar to the FDR plot and thus is not shown.

Both plots showed that extBIC always has higher F1 scores than BIC for all methods across different number of genes (Figure S1). The difference between BIC, extBIC and oracal is substantial for the NS methods. In contrast the difference is much smaller for the space methods with the extBIC F1 score very close to the oracle F1 score. Like F1, the FDR for extBIC is much smaller than BIC across all methods with varying m (Figure S2).

In summary, extBIC performs better than BIC and its performance is close to Oracle for the space approach. In the following simulation we thus use extBIC (Chen and Chen, 2008, 2012) to choose the tuning parameter in comparing the performance between different GGN methods.

S3 Simulation Studies

In this section, we present Monte Carlo simulation to evaluate the performance of the `space-log`, `space-lasso`, `NS-log`, and `NS-lasso`. Without loss of generality, we simulated data sets with $n = 400$ individuals. We considered different number of genes $m = 100, 200, 300$, respectively. Following Ha *et al.* (2016), we studied two types of graphs: the traditional random graphs (ER model) where all the genes have the same expected number of neighbors (Kalisch and Bühlmann, 2007, Rényi and Erdos, 1960), and hubs graphs where a few genes may have a large number of neighbors (BA model), where BA model is more frequently observed in gene networks (Barabási and Albert, 1999). Figure S3 and S4 show the results under the BA model with low number of connections (e=1) and high number of connections (e=2), respectively. Figure S5 and S6 show the results under the ER model with low and high numbers of connections, respectively. Figure S7 compared the running time in seconds. Figure S3-S7 are cited in main article.

Table S1: Simulation Settings

m	n	p_E (ER)	e (BA)
400	100	1/100, 2/100	1,2
400	200	1/200, 2/200	1,2
400	300	1/300, 2/300	1,2

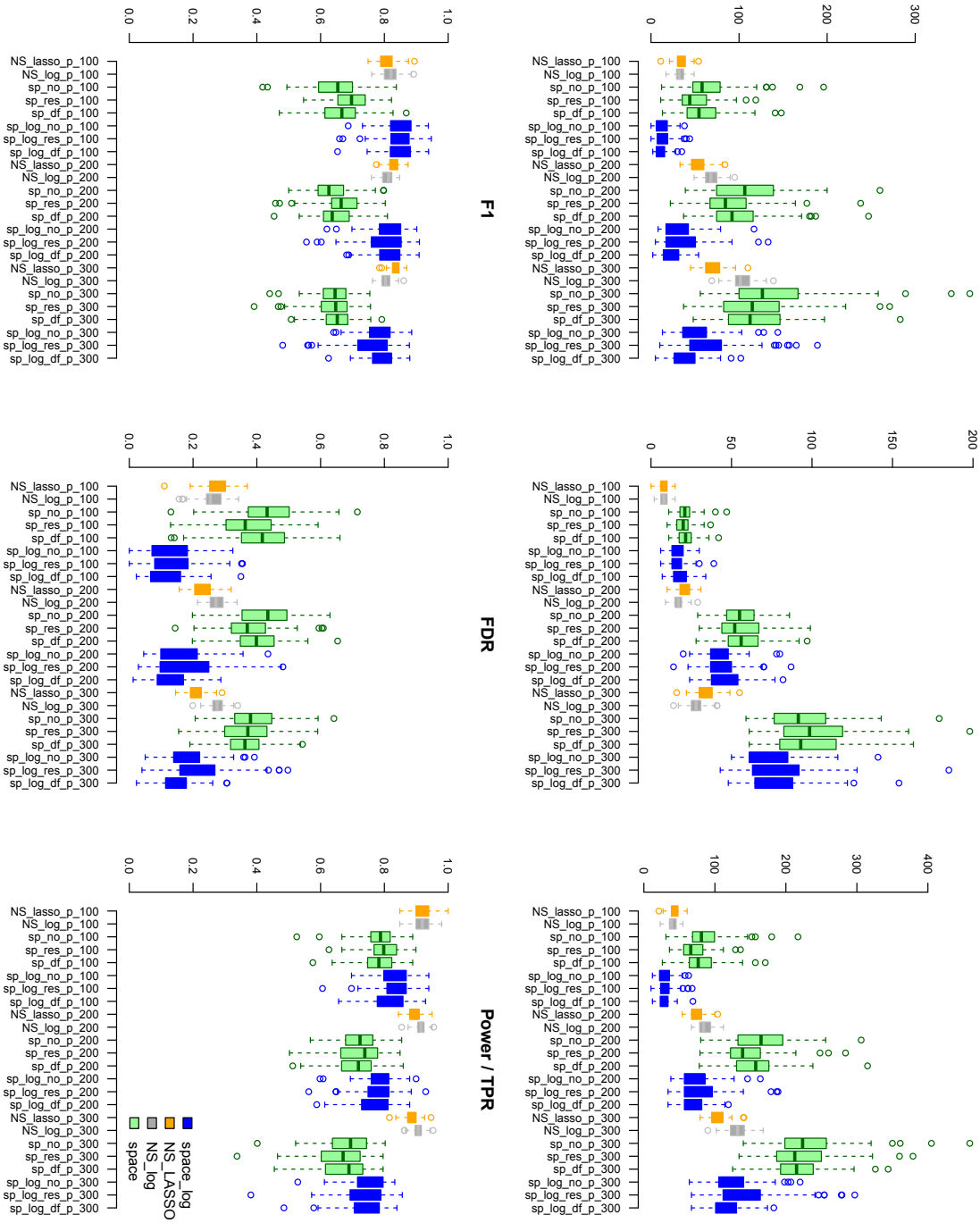


Figure S3: BA with 100/200/300 genes and e=1

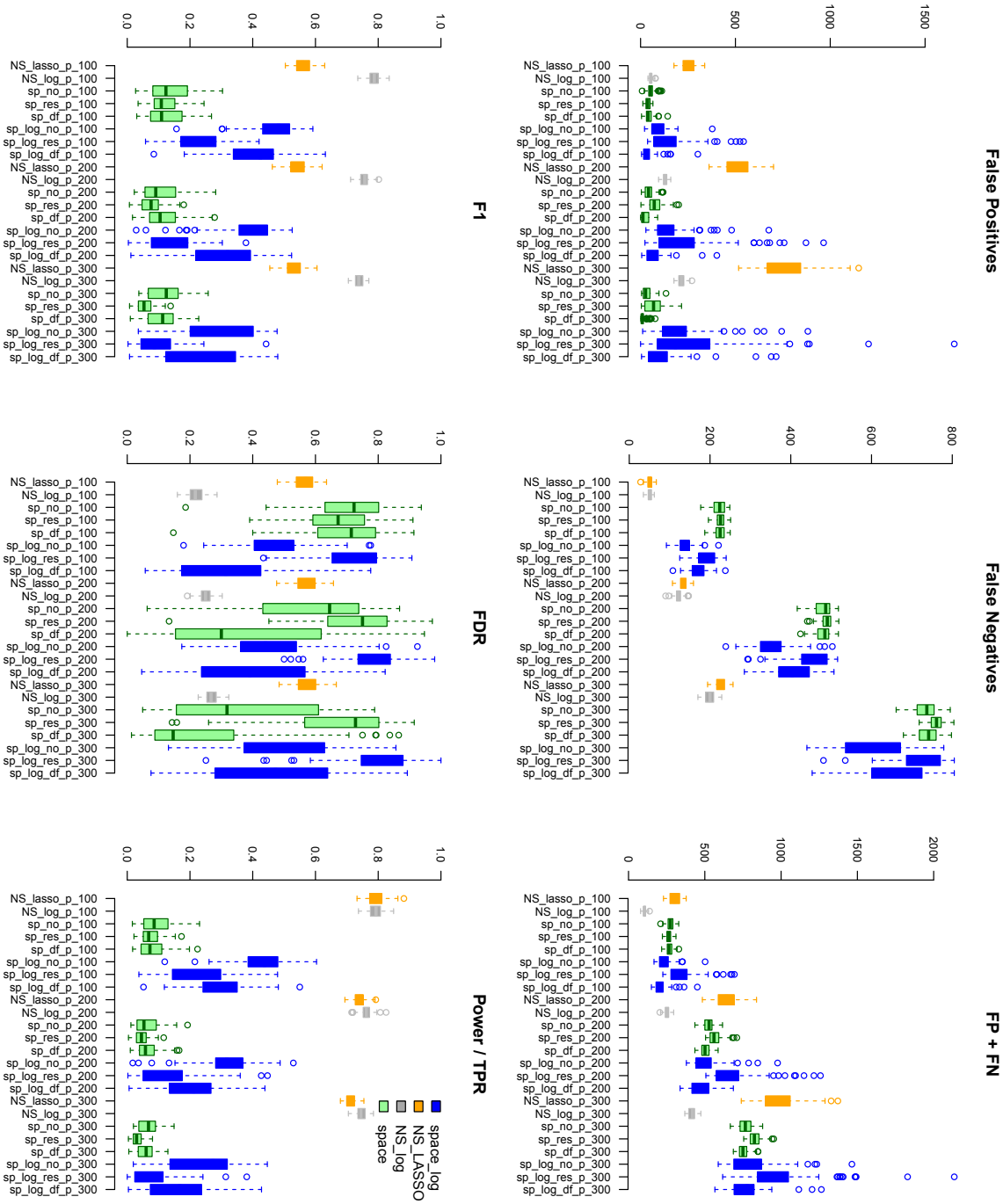


Figure S4: BA with 100/200/300 genes and e=2

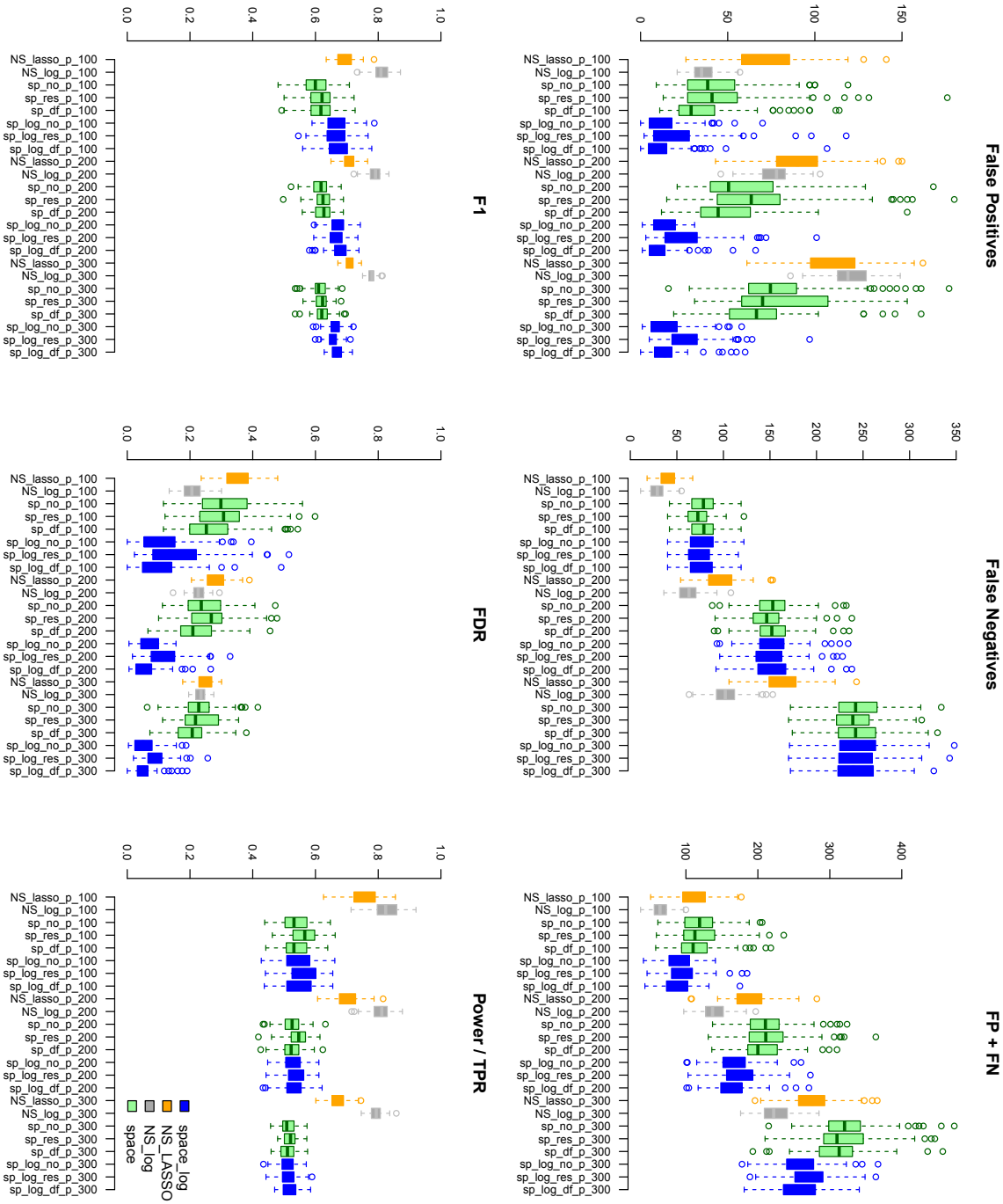


Figure S5: ER with 100/200/300 genes and $p_E = 1/m$

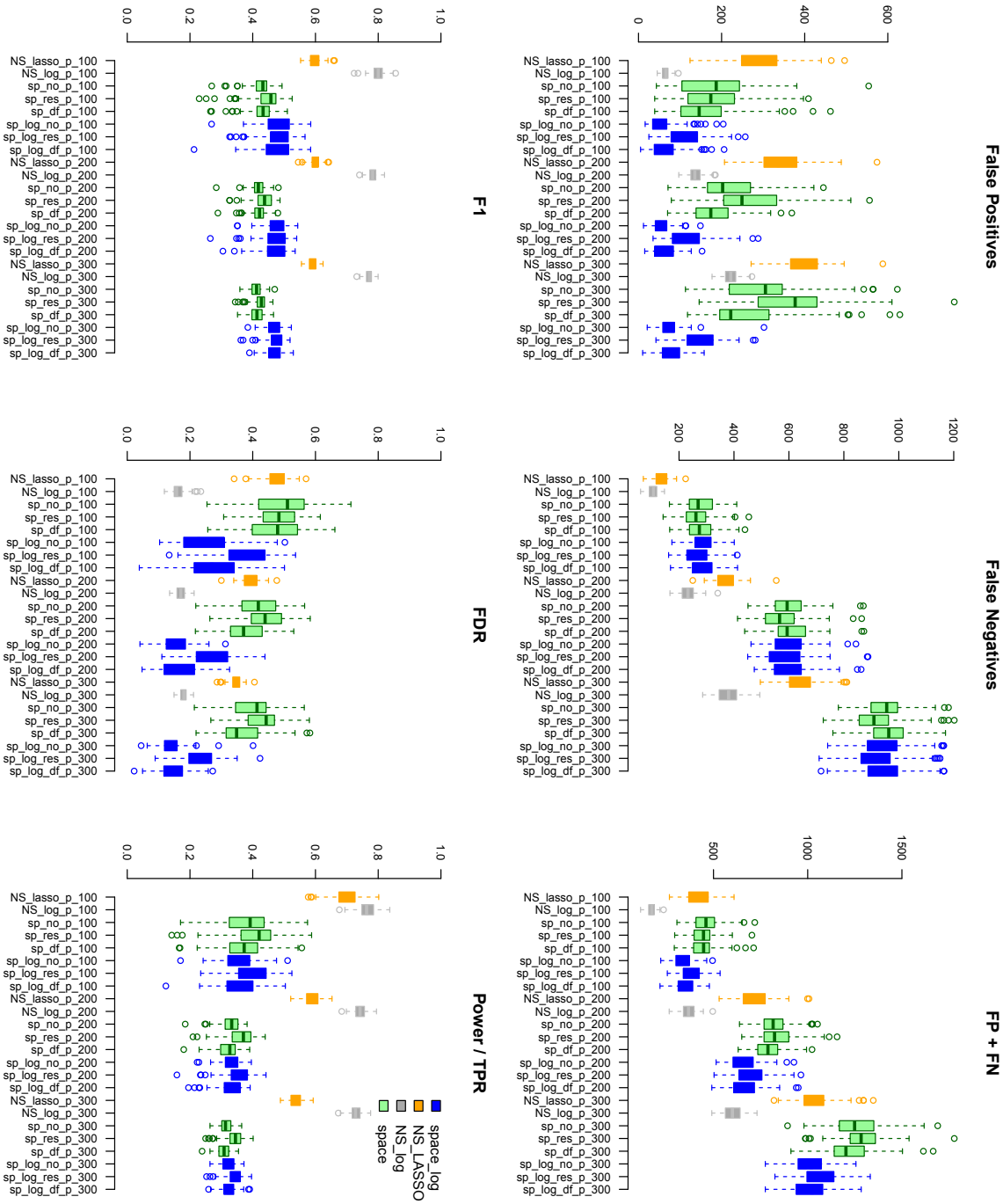


Figure S6: ER with 100/200/300 genes and $p_E = 2/m$

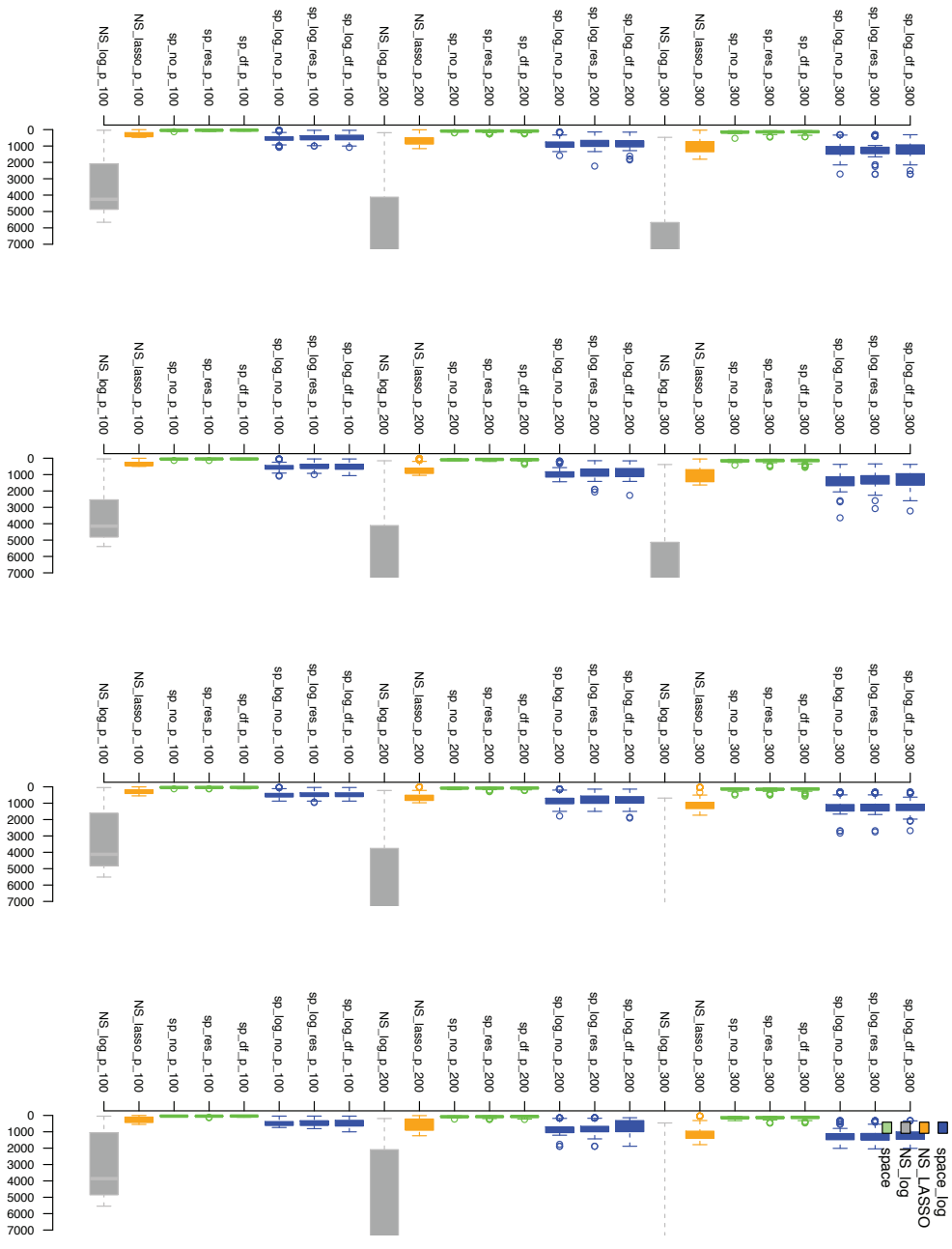


Figure S7: Boxplot on the computational times

S4 Application to GTEx and TCGA data

S4.1 TCGA data

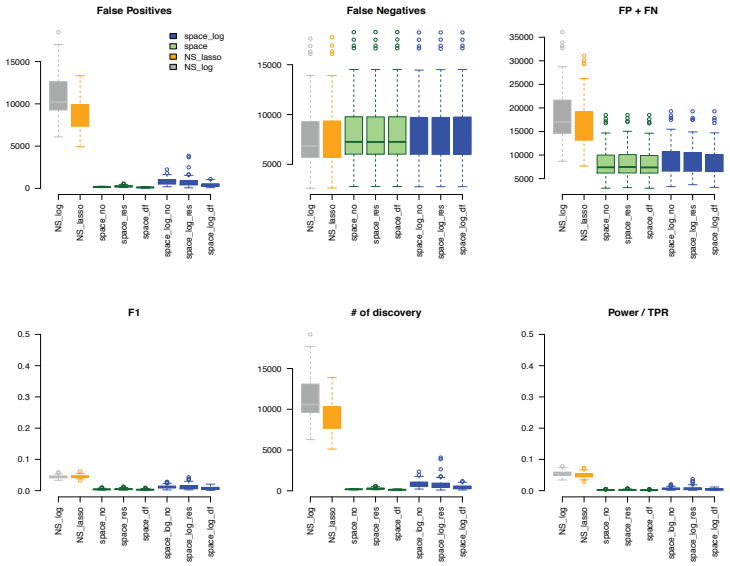


Figure S8: TCGA data analysis with 71 larger gene sets (p>n)

To further understand whether our proposed methods can better handle high dimensional data with larger number of genes, we generated k=71 larger network by combining gene sets with overlapping genes. The sizes of this new larger gene sets list ranged from 476 to 1253 genes, which is larger than the sample size n=450 (m > n). We used exact the same procedure in Application to TCGA data section in main paper, and calculated similar metrics (FP, FN, FP+FN, number of total discovery, F1 score, and true positive rate (Figure S8)). Comparing with TCGA data with smaller gene sets (Figure 4 and 5), we observed that the NS-based approach with both lasso and log penalty discovered much more false positive edges than space-based approaches for this new larger gene sets setting, which is consistent with our finding in simulation that NS-based approach has more false positive in general.

S4.2 Source codes

The source code `space-log` is available on github https://github.com/wuqian77/SpaceLog.

The source code `NS-log` and `NS-lasso` is available on github https://github.com/Sun-lab/penalized_estimation/,

and the existing methods (`space-lasso` is available on R CRAN https://cran.r-project.org/web/packages/space/index.html.

Acknowledgements

This research is supported by NIH grants R01 CA189532.

References

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *science*, **286**(5439), 509–512.

Chen, J. and Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, **95**(3), 759–771.

Chen, J. and Chen, Z. (2012). Extended bic for small-n-large-p sparse glm. *Statistica Sinica*, pages 555–574.

Chen, T.-H., Sun, W., and Fine, J. P. (2016). Designing penalty functions in high dimensional problems: The role of tuning parameters. *Electronic journal of statistics*, **10**(2), 2312.

Fan, J., Feng, Y., and Wu, Y. (2009). Network exploration via the adaptive lasso and scad penalties. *The annals of applied statistics*, **3**(2), 521.

Ha, M. J., Sun, W., and Xie, J. (2016). Penpc: A two-step approach to estimate the skeletons of high-dimensional directed acyclic graphs. *Biometrics*, **72**(1), 146–155.

Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, **8**(Mar), 613–636.

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**(486), 735–746.

Rényi, P. E.-A. and Erdos, P. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci. A*, **5**, 17–61.

Wang, H., Li, B., and Leng, C. (2009). Shrinkage tuning parameter selection with a diverging number of parameters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **71**(3), 671–683.

Zou, H., Hastie, T., Tibshirani, R., *et al.* (2007). On the "degrees of freedom" of the lasso. *The Annals of Statistics*, **35**(5), 2173–2192.