Identifying differential networks based on multi-platform gene expression data

Le Ou-Yang, Hong Yan and Xiao-Fei Zhang

1 Supplementary Figures

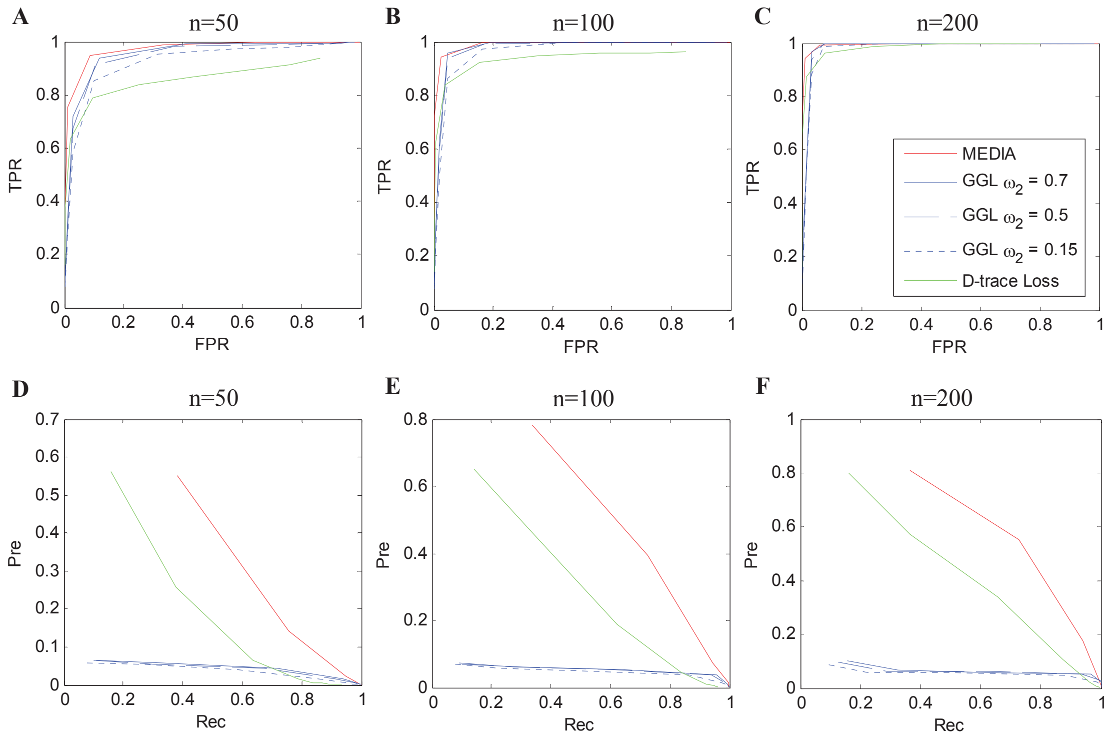

Figure S1: Performance of different methods on simulated data (Simulation 2) with $p = 100$, $K = 3$ and $n_x = n_y = 50, 100, 200$. (A-C) are ROC curves. (D-F) are precision-recall curves. Red line: MEDIA; blue line: grouped graphical lasso (GGL); green line: D-trace loss estimator. For GGL, each colored curve corresponds to a fixed value of ω_2, with the value of ω_1 varied. Each curve is averaged over 100 random generations of the data.

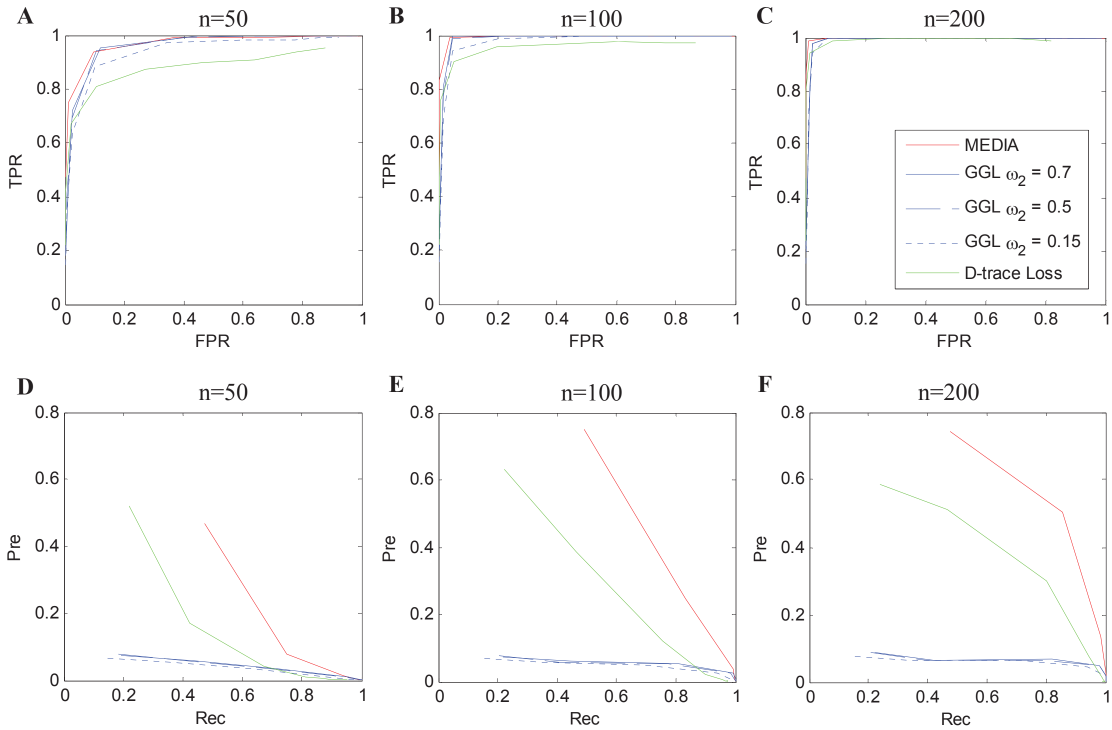

Figure S2: Performance of different methods on simulated data (Simulation 3) with $p = 100$, $K = 3$ and $n_x = n_y = 50, 100, 200$. (A-C) are ROC curves. (D-F) are precision-recall curves. Red line: MEDIA; blue line: grouped graphical lasso (GGL); green line: D-trace loss estimator. For GGL, each colored curve corresponds to a fixed value of ω_2, with the value of ω_1 varied. Each curve is averaged over 100 random generations of the data.

2 Supplementary Text

2.1 Algorithm for parameter estimation

In this section, we solve the optimization problem in the main text by using an alternating direction method of multipliers (ADMM) [1]. We first introduce K auxiliary matrices $\{\tilde{\triangle}\} = \{\tilde{\triangle}_k\}_{k=1}^K$ and rewrite the objective function as

$$\begin{cases} \min\limits_{\substack{\triangle_k=\triangle_k^T \\ k=1,\ldots,K}} L_D\left(\{\triangle\}, \{\hat{\Sigma}_X\}, \{\hat{\Sigma}_Y\}\right) + \lambda P\left(\{\tilde{\triangle}\}\right) \\ \text{subject to} \quad \triangle_k = \tilde{\triangle}_k, \quad k = 1, \ldots, K. \end{cases} \tag{1}$$

From (1), we consider the following augmented Lagrangian function:

$$\begin{aligned} L\left(\{\triangle\}, \{\tilde{\triangle}\}, \{A\}\right) &= L_D\left(\{\triangle\}, \{\hat{\Sigma}_X\}, \{\hat{\Sigma}_Y\}\right) + \lambda P\left(\{\tilde{\triangle}\}\right) \\ &+ \sum_{k=1}^K \left\langle A_k, \triangle_k - \tilde{\triangle}_k \right\rangle + (\rho/2) \sum_{k=1}^K \|\triangle_k - \tilde{\triangle}_k\|_F^2. \end{aligned} \tag{2}$$

where $\{A\} = A_1, \ldots, A_K$ are dual variables and ρ serves as a penalty parameter. Given the solution $\left(\{\triangle^t\}, \{\tilde{\triangle}^t\}, \{A^t\}\right)$ at the t-th step ($t = 1, 2, \ldots$), we update $\left(\{\triangle^{t+1}\}, \{\tilde{\triangle}^{t+1}\}, \{A^{t+1}\}\right)$ as follows:

$$\{\triangle_k^{t+1}\}_{k=1}^K = \mathop{\arg\min}_{\substack{\triangle_k=\triangle_k^T \\ k=1,\ldots,K}} L\left(\{\triangle\}, \{\tilde{\triangle}^t\}, \{A^t\}\right) \tag{3}$$

$$\{\tilde{\triangle}_k^{t+1}\}_{k=1}^K = \mathop{\arg\min}_{\substack{\tilde{\triangle}_k=\tilde{\triangle}_k^T \\ k=1,\ldots,K}} L\left(\{\triangle^{t+1}\}, \{\tilde{\triangle}\}, \{A^t\}\right) \tag{4}$$

$$A_k^{t+1} = A_k^t + \rho(\triangle_k^{t+1} - \tilde{\triangle}_k^{t+1}), k = 1, \ldots, K. \tag{5}$$

In particular, updating $\triangle_k$, $k = 1, \ldots, K$ by minimizing $L\left(\{\triangle\}, \{\tilde{\triangle}^t\}, \{A^t\}\right)$ is equivalent to solving the following optimization problem:

$$\triangle_k^{t+1} = \mathop{\arg\min}_{\triangle_k=\triangle_k^T} \frac{1}{2}\left\langle \triangle_k^2, (\hat{\Sigma}_X^k\hat{\Sigma}_Y^k + \hat{\Sigma}_Y^k\hat{\Sigma}_X^k)/2 + \rho I \right\rangle - \left\langle \triangle_k, \hat{\Sigma}_X^k - \hat{\Sigma}_Y^k + \rho\tilde{\triangle}_k^t - A_k^t \right\rangle. \tag{6}$$

where I is a $p \times p$ identity matrix. According to Theorem 1 of [3], the explicit solution to (6) is given by $\triangle_k^{t+1} = G\left((\hat{\Sigma}_X^k\hat{\Sigma}_Y^k + \hat{\Sigma}_Y^k\hat{\Sigma}_X^k)/2 + \rho I, \hat{\Sigma}_X^k - \hat{\Sigma}_Y^k + \rho\tilde{\triangle}_k^t - A_k^t\right)$, where $G(\Psi, \Phi) = U_\Psi\{(U_\Psi^T \Phi U_\Psi) \circ C\}U_\Psi^T$, and $\Psi = U_\Psi \Sigma_\Psi U_\Psi^T$ is the eigenvalue decomposition of Ψ, with ordered eigenvalues $\sigma_1 \geq \cdots \geq \sigma_p$, $\circ$ denotes the Hadamard product of matrices and $C_{ij} = 2/(\sigma_i + \sigma_j)$.

Updating $\tilde{\triangle}_k, k = 1, \ldots, K$ by minimizing $L\left(\{\triangle^{t+1}\}, \{\tilde{\triangle}\}, \{A^t\}\right)$ is equivalent to solving the following optimization problem:

$$\{\tilde{\triangle}_k^{t+1}\}_{k=1}^K = \underset{\substack{\tilde{\triangle}_k = \tilde{\triangle}_k^T \\ k=1,\ldots,K}}{\arg\min} \frac{\rho}{2}\sum_{k=1}^K \|\tilde{\triangle}_k - \triangle_k^{t+1} - A_k^t/\rho\|_F^2 + \lambda \sum_{i \neq j}\left(\sum_{k=1}^K (\tilde{\triangle}_k)_{ij}^2\right)^{\frac{1}{2}}. \tag{7}$$

Similar to [2], for all $i = 1, \ldots, p$ and $k = 1, \ldots, K$, the solution to problem (7) has $(\tilde{\triangle}_k)_{ii} = (\triangle_k^{t+1} + A_k^t/\rho)_{ii}$. While the off-diagonal elements take the form

$$(\tilde{\triangle}_k^{t+1})_{ij} = (B_k)_{ij}\left(1 - \frac{\lambda}{\rho\{\sum\limits_{k=1}^K (B_k)_{ij}^2\}^{\frac{1}{2}}}\right)_+ \tag{8}$$

where $B_k = \triangle_k^{t+1} + A_k^t/\rho$. Based on the analysis above, the complete ADMM algorithm is summarized in Algorithm 1. This algorithm can be accelerated by adaptively changing ρ. We set $\rho = 0.1$ and increase it iteratively by scaling it to $u\rho$. Here, we set $u = 1.05$. In the implementation of the algorithm, the convergence condition is

$$\sum_{k=1}^K \|\triangle_k^{t+1} - \triangle_k^t\| < \varepsilon \cdot \max(1, \sum_{k=1}^K \|\triangle_k^t\|, \sum_{k=1}^K \|\triangle_k^{t+1}\|). \tag{9}$$

where ε is a tolerance which is set to 10^{-3} in our experiments.

Algorithm 1 ADMM algorithm for the multi-view differential network analysis model.

Input:
sample covariance matrices $\{\hat{\Sigma}_X\}$ and $\{\hat{\Sigma}_Y\}$, parameter λ.
Initialize:
$t = 0$, $A_k^0 = 0$, $\triangle_k^0 = \tilde{\triangle}_k^0 = I$, $k = 1, \ldots, K$, $\rho = 0.1$, $u = 1.05$, $max_\rho = 10^{10}$, $\varepsilon = 10^{-3}$.
while (not converged) **do**

1: $t = t + 1$;

2: fix the others and update $\triangle_k$, $k = 1, \ldots, K$ by $\triangle_k^{t+1} = G\left((\hat{\Sigma}_X^k\hat{\Sigma}_Y^k + \hat{\Sigma}_Y^k\hat{\Sigma}_X^k)/2 + \rho I, \hat{\Sigma}_X^k - \hat{\Sigma}_Y^k + \rho\tilde{\triangle}_k^t - A_k^t\right)$

3: fix the others and update $\tilde{\triangle}_k$, $k = 1, \ldots, K$ by $(\tilde{\triangle}_k^{t+1})_{ij} = (B_k)_{ij}\left(1 - \frac{\lambda}{\rho\{\sum_{k=1}^K (B_k)_{ij}^2\}^{\frac{1}{2}}}\right)_+$;

4: update the multipliers $A_k^{t+1} = A_k^t + \rho(\triangle_k^{t+1} - \tilde{\triangle}_k^{t+1}), k = 1, \ldots, K$

5: update the parameter ρ by $\rho = min(u\rho, max_\rho)$

6: check the convergence conditions $\sum_{k=1}^K \|\triangle_k^{t+1} - \triangle_k^t\| < \varepsilon \cdot \max(1, \sum_{k=1}^K \|\triangle_k^t\|, \sum_{k=1}^K \|\triangle_k^{t+1}\|)$.

end while

References

[1] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[2] Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

[3] TENG ZHANG and HUI ZOU. Sparse precision matrix estimation via lasso penalized d-trace loss. *Biometrika*, 101(1):103–120, 2014.