Differential network analysis from cross-platform gene expression data:
Supplementary Information

Xiao-Fei Zhang, Le Ou-Yang, Xing-Ming Zhao, and Hong Yan

Contents

1 Supplementary Figures 2

2 Supplementary Text 7

2.1 Brief review of ADMM algorithms 7
2.2 ADMM algorithm for fused graphical lasso with weighted $\ell_1$ penalty 7
  2.2.1 $\Theta^{k_1}$ and $\Theta^{k_2}$ update 8
  2.2.2 $Z^{k_1}$ and $Z^{k_2}$ update 8
  2.2.3 Complete ADMM algorithm 8
  2.2.4 Stop criteria 8
  2.2.5 Varying penalty parameter 9
2.3 Complete algorithm of the TDJGL model 9
2.4 Model selection 10
2.5 Criteria for platinum response groups 11
2.6 Comparison with other graphical lasso models on the ovarian cancer data 11
1 Supplementary Figures

Figure S1: Examples for three types of networks used to generate simulated data sets. (a) Erdős-Rényi network, (b) scale-free network, and (c) community network.

Figure S2: Performance of the compared methods on Erdos-Renyi network with $p = 100$, $K = 3$, $\tau = 10\%$ and (a) $n = 50$, (b) $n = 100$, (c) $n = 200$. Each colored line corresponds to a fixed value of $\lambda_2$ ($\omega_2$ for GGL), as $\lambda_1$ ($\omega_1$ for GGL) is varied. Variables corresponding to the axes are explained in Table 1 in the main text. Results are averaged over 100 random generations of the data.
Figure S3: Performance of the compared methods on community network with $p = 100$, $K = 3$, $\tau = 10\%$ and (a) $n = 50$, (b) $n = 100$, (c) $n = 200$. Each colored line corresponds to a fixed value of $\lambda_2$ ($\omega_2$ for GGL), as $\lambda_1$ ($\omega_1$ for GGL) is varied. Variables corresponding to the axes are explained in Table 1 in the main text. Results are averaged over 100 random generations of the data.
Figure S4: Performance of the compared models on scale-free network with $p = 100$, $K = 3$, $\tau = 20\%$ and (a) $n = 50$, (b) $n = 100$, (c) $n = 200$. Each colored line corresponds to a fixed value of $\lambda_2$ (\(\omega_2\) for GGL), as $\lambda_1$ (\(\omega_1\) for GGL) is varied. Variables corresponding to the axes are explained in Table 1 in the main text. Results are averaged over 100 random generations of the data.
Figure S5: Performance of the compared models on scale-free network with $p = 100$, $K = 3$, $\tau = 50\%$ and (a) $n = 50$, (b) $n = 100$, (c) $n = 200$. Each colored line corresponds to a fixed value of $\lambda_2$ ($\omega_2$ for GGL), as $\lambda_1$ ($\omega_1$ for GGL) is varied. Variables corresponding to the axes are explained in Table 1 in the main text. Results are averaged over 100 random generations of the data.
Figure S6: Overlaps between the edges (and differential edges) detected by TDJGL from the three platforms for (a) platinum-resistant tumors, (b) platinum-sensitive tumors and (c) differential networks.
2 Supplementary Text

2.1 Brief review of ADMM algorithms

In this section, we briefly review the standard alternating direction method of multipliers (ADMM) algorithms [1]. ADMM is a technique for solving optimization problem in the following form [2]:

\[
\min_{X} \; f(X) + g(X)
\]

subject to \( X \in \mathcal{X} \).

(1)

ADMM is attractive when the proximal operator of \( f(X) + g(X) \) cannot be easily obtained, but the proximal operator of \( f(X) \) and the proximal operator of \( g(X) \) can be easily computed. The approach consists of the following three steps [2]:

1. Rewrite the problem (1) as

\[
\min_{X, Z} \; f(X) + g(Z)
\]

subject to \( X \in \mathcal{X}, X = Z \),

(2)

where functions \( f \) and \( g \) are decoupled by introducing a new optimization variable, \( Z \).

2. Form the augmented Lagrangian

\[
L_\rho(X, Z, \Lambda) = f(X) + g(Z) + \langle \Lambda, X - Z \rangle + \frac{\rho}{2} \|X - Z\|_F^2,
\]

where \( \Lambda \) is the Lagrange multiplier and \( \rho > 0 \) is a penalty parameter.

3. Iterate the following two steps until convergence

(a) Update each primal variable in turn by minimizing the augmented Lagrangian (3) with respect to that variable, while holding all other variables. The update in the \( t \)th iteration are as follows:

\[
X^{t+1} \leftarrow \arg \min_{X \in \mathcal{X}} L_\rho(X, Z^t, \Lambda^t),
\]

\[
Z^{t+1} \leftarrow \arg \min_{Z} L_\rho(X^t, Z, \Lambda^t).
\]

(b) Update the dual variable using a dual-ascent update:

\[
\Lambda^{t+1} \leftarrow \Lambda^t + \rho(X^{t+1} - Z^{t+1}).
\]

2.2 ADMM algorithm for fused graphical lasso with weighted \( \ell_1 \) penalty

Following in the method of [3], we use the ADMM algorithm to solve the fused graphical lasso with weighted \( \ell_1 \) penalty (problem (5) in the main text)

\[
\min_{\Theta^{k_1}, \Theta^{k_2} \in S_{++}^{2n}, Z^{k_1}, Z^{k_2}} \sum_{c=1}^{2} n_c \left( \text{tr} (S^{kc} \Theta^{kc}) - \log \det (\Theta^{kc}) \right) + \lambda_1 \sum_{c=1}^{2} \sum_{i \neq j} \omega_{ij} |\theta_{ij}^{kc}| + \lambda_2 \sum_{i,j} \psi_{ij} |\theta_{ij}^{k1} - \theta_{ij}^{k2}|.
\]

(4)

To solve the problem (4), we reformulate it by introducing new variables \( Z^{k1} \) and \( Z^{k2} \), so as to decouple some of the terms in the objective function that are difficult to optimize jointly:

\[
\min_{\Theta^{k_1}, \Theta^{k_2} \in S_{++}^{2n}, Z^{k_1}, Z^{k_2}} \sum_{c=1}^{2} n_c \left( \text{tr} (S^{kc} \Theta^{kc}) - \log \det (\Theta^{kc}) \right) + \lambda_1 \sum_{c=1}^{2} \sum_{i \neq j} \omega_{ij} |\theta_{ij}^{kc}| + \lambda_2 \sum_{i,j} \psi_{ij} |\theta_{ij}^{k1} - \theta_{ij}^{k2}|.
\]

subject to

\[
\Theta^{k1} = Z^{k1}, \Theta^{k2} = Z^{k2}.
\]

(5)

The augmented Lagrangian to (5) is given by

\[
L_\rho(\Theta^{k_1}, \Theta^{k_2}, Z^{k_1}, Z^{k_2}, U^{k_1}, U^{k_2}) = \sum_{c=1}^{2} n_c \left( \text{tr} (S^{kc} \Theta^{kc}) - \log \det (\Theta^{kc}) \right) + \lambda_1 \sum_{c=1}^{2} \sum_{i \neq j} \omega_{ij} |\theta_{ij}^{kc}| + \lambda_2 \sum_{i,j} \psi_{ij} |\theta_{ij}^{k1} - \theta_{ij}^{k2}| + \langle U^{k1}, \Theta^{k1} - Z^{k1} \rangle + \langle U^{k2}, \Theta^{k2} - Z^{k2} \rangle + \frac{\rho}{2} \|\Theta^{k1} - Z^{k1}\|_F^2 + \frac{\rho}{2} \|\Theta^{k2} - Z^{k2}\|_F^2.
\]

(6)

where \( U^{k1} \) and \( U^{k2} \) are dual variables and \( \rho \) serves as the penalty parameter.

The ADMM algorithm updates each primal variable while holding the other variables fixed and then updates the dual variables using a dual-ascent update rule. We now derive the update rules for the variables.
2.2.1 \( \Theta^{k1} \) and \( \Theta^{k2} \) update

Before introducing the update rules for \( \Theta^{k1} \) and \( \Theta^{k2} \), we first define the Expand operator:

\[
\text{Expand} (A, \rho, n) = \arg \min_{\Theta \in S_{n+}^+} -n \log \det (\Theta) + \frac{\rho}{2} \| \Theta - A \|^2 = \frac{1}{2} U \left( D + \sqrt{D^2 + \frac{4n}{\rho}} I \right) U^T, \tag{7}
\]

where \( U D U^T \) is the eigenvalue decomposition of a symmetric matrix \( A \). The Expand operator has been used to solve the graphical lasso problem in previous studies [2, 4, 3].

Note that

\[
\Theta^{k1} = \arg \min_{\Theta^{k1} \in S_{n+}^+} L_\rho \left( \Theta^{k1}, \Theta^{k2}, Z^{k1}, Z^{k2}, U^{k1}, U^{k2} \right)
= \arg \min_{\Theta^{k1} \in S_{n+}^+} -n \log \det (\Theta^{k1}) + \frac{\rho}{2} \left\| \Theta^{k1} - \left( Z^{k1} - \frac{1}{\rho} (n_1 S^{k1} + U^{k1}) \right) \right\|^2_F
\tag{8}
\]

Now it follows from the definition of the Expand operator that

\[
\Theta^{k1} \leftarrow \text{Expand} \left( Z^{k1} - \frac{1}{\rho} (n_1 S^{k1} + U^{k1}) , \rho, n_1 \right).
\tag{9}
\]

The update for \( \Theta^{k2} \) can be derived in a similar way:

\[
\Theta^{k2} \leftarrow \text{Expand} \left( Z^{k2} - \frac{1}{\rho} (n_2 S^{k2} + U^{k2}) , \rho, n_2 \right).
\tag{10}
\]

2.2.2 \( Z^{k1} \) and \( Z^{k2} \) update

Minimizing augmented Lagrangian (6) with respect to \( Z^{k1} \) and \( Z^{k2} \) can be written as follows:

\[
\{Z^{k1}, Z^{k2}\} = \arg \min_{Z^{k1}, Z^{k2}} L_\rho \left( \Theta^{k1}, \Theta^{k2}, Z^{k1}, Z^{k2}, U^{k1}, U^{k2} \right)
= \arg \min_{Z^{k1}, Z^{k2}} \frac{\rho}{2} \sum_{c=1}^{2} \left\| Z^{k_c} - \left( \Theta^{k_c} + \frac{U^{k_c}}{\rho} \right) \right\|^2_F + \lambda_1 \sum_{c=1}^{2} \sum_{i \neq j} \omega_{ij} |z_{ij}^{k_c} - z_{ij}^{k_2}| + \lambda_2 \sum_{i,j} \psi_{ij} |z_{ij}^{k1} - z_{ij}^{k2}| \tag{11}
\]

Now (11) is completely separable with respect to each pair of matrix elements \((i, j)\). That is, we can solve for each \((i, j)\):

\[
\{z_{ij}^{k1}, z_{ij}^{k2}\} = \arg \min_{z_{ij}^{k1}, z_{ij}^{k2}} \frac{\rho}{2} \sum_{c=1}^{2} \left( z_{ij}^{k_c} - a_{ij}^{k_c} \right)^2 + \lambda_1 \sum_{c=1}^{2} \omega_{ij} |z_{ij}^{k_c} - z_{ij}^{k_2}| + \lambda_2 \psi_{ij} |z_{ij}^{k1} - z_{ij}^{k2}|,
\tag{12}
\]

where \( a_{ij}^{k_c} = z_{ij}^{k_c} + \frac{\eta_{ij}}{\rho} \). This is a special case of the fused lasso signal approximator [3, 5] and it has a very simple closed form solution. When \( \lambda_1 = 0 \), the solution to (12) takes the form

\[
(z_{ij}^{k1}, z_{ij}^{k2}) = \begin{cases} (a_{ij}^{k1} - \lambda_2 \psi_{ij} / \rho, a_{ij}^{k2} + \lambda_2 \psi_{ij} / \rho) & \text{if } a_{ij}^{k1} > a_{ij}^{k2} + 2 \lambda_2 \psi_{ij} / \rho \\ (a_{ij}^{k1} + \lambda_2 \psi_{ij} / \rho, a_{ij}^{k2} - \lambda_2 \psi_{ij} / \rho) & \text{if } a_{ij}^{k1} > a_{ij}^{k2} + 2 \lambda_2 \psi_{ij} / \rho \\ \left( a_{ij}^{k1}, a_{ij}^{k2} \right) & \text{if } |a_{ij}^{k1} - a_{ij}^{k2}| \leq 2 \lambda_2 \psi_{ij} / \rho \end{cases} \tag{13}
\]

When \( \lambda_1 > 0 \), the solution to (12) can be obtained through soft-thresholding (13) by \( \lambda_1 \omega_{ij} \). Here the soft-thresholding is defined as \( S(x, c) = \text{sign}(x) \left( |x| - c \right)_+ \), where \( a_+ = \max(a, 0) \).

2.2.3 Complete ADMM algorithm

Based on the augmented Lagrangian, the complete ADMM algorithm for (4) is given in Algorithm 1. We find it is useful in practice to vary the value of penalty parameter \( \rho \) for each iteration. Therefore, we update \( \rho \) in each iteration based on the primal residues and dual residues. We present the update rule for \( \rho \) at the end of this section. Please note that according to the ADMM algorithm, the estimates \( \hat{\Theta}^{k1} \) and \( \hat{\Theta}^{k1} \) are not exactly sparse, while \( \hat{Z}^{k1} \) and \( \hat{Z}^{k1} \) are sparse which are obtained through soft-thresholding.

2.2.4 Stop criteria

Let \( \Theta^{k_c}_t, Z^{k_c}_t \) and \( U^{k_c}_t \) denote the estimates at the \( t \)th iteration. The primal residual [1] is defined as

\[
P_t = \begin{bmatrix} \Theta^{k1}_t \\ \Theta^{k2}_t \end{bmatrix} - \begin{bmatrix} Z^{k1}_t \\ Z^{k2}_t \end{bmatrix}, \tag{14}
\]
We consider the process converges if both primal residual and dual residual are sufficiently small \cite{1}. More specially, we posed (19) into and the dual residual \cite{1} is defined as

\[ \frac{1}{2} \rho (n_2 S^{k_2} + U^{k_2}) + \rho, n_2) ; \]

\[ \text{while Not converged do} \]

1. \[ \Theta^{k_1} \leftarrow \text{Expand} \left( Z^{k_1} = \frac{1}{\rho} (n_1 S^{k_1} + U^{k_1}), \rho, n_1) ; \]

2. \[ \Theta^{k_2} \leftarrow \text{Expand} \left( Z^{k_2} = \frac{1}{\rho} (n_2 S^{k_2} + U^{k_2}), \rho, n_2) ; \]

3. Update \( Z^{k_1} \) and \( Z^{k_2} \) through solving problem (11);

4. Update \( U^{k_1} \leftarrow U^{k_1} + \rho * (\Theta^{k_1} - Z^{k_1}); \]

5. Update \( U^{k_2} \leftarrow U^{k_2} + \rho * (\Theta^{k_2} - Z^{k_2}); \]

6. Update \( \rho \) according to (18).

\[ \text{Output } \Theta^{k_1}, \Theta^{k_1}, Z^{k_1} \text{ and } Z^{k_1} \]

and the dual residual \cite{1} is defined as

\[ D_t = \rho \left( \left[ Z^{k_1}_{t-1} \right] - \left[ Z^{k_2}_{t-1} \right] \right). \]

We consider the process converges if both primal residual and dual residual are sufficiently small \cite{1}. More specially, we introduce small positive constants \( \varepsilon^{\text{abs}} \) and \( \varepsilon^{\text{rel}} \), and declare \( P_t \) and \( D_t \) small if

\[ \| P_t \|_F \leq \sqrt{2} p\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}} \max \left\{ \left\| \Theta^{k_1} \right\|_F, \left\| Z^{k_1} \right\|_F \right\}, \]

\[ \| D_t \|_F \leq \sqrt{2} p\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}} \left\| U^{k_1} \right\|_F, \]

We set \( \varepsilon^{\text{abs}} = 10^{-3} \) and \( \varepsilon^{\text{rel}} = 10^{-3} \) as suggested by Boyd et al. \cite{1}. The choice of the value of \( \varepsilon^{\text{abs}} \) depends on the scale of the variable values.

2.2.5 Varying penalty parameter

In practice, it is useful to use different penalty parameter \( \rho \) for each iteration, which might improve the convergence as well as make performance less dependent on the initial value of penalty parameter \cite{1}. Therefore, we update the value of \( \rho \) in the \( t \)th iteration according to the primal residuals and dual residuals:

\[ \rho \left\{ \begin{array}{ll} \tau^{\text{incr}} \rho & \text{if } \| P_t \|_F > \mu \| D_t \|_F \\ \rho/\tau^{\text{decr}} & \text{if } \| D_t \|_F > \mu \| P_t \|_F \\ \rho & \text{otherwise} \end{array} \right. \]

(18)

where \( \mu > 1, \tau^{\text{incr}} > 1 \) and \( \tau^{\text{decr}} > 1 \) are the adaptation parameters. We set \( \mu = 10, \tau^{\text{incr}} = 2 \) and \( \tau^{\text{decr}} = 2 \) according to \cite{1}.

2.3 Complete algorithm of the TDJGL model

By taking Equation (4) into Equation (3) in the main text, the TDJGL model can be written as follows:

\[ \min_{\Theta^{k_1}} \sum_{k=1}^{K} \sum_{c=1}^{n_c} \left( \text{tr} \left( S^{k_c} \Theta^{k_c} \right) - \log \det \left( \Theta^{k_c} \right) \right) + \lambda_1 \sum_{i,j} \sqrt{\sum_{k=1}^{K} \sum_{c=1}^{n_c} | \theta^{k_c} |} + \lambda_2 \sum_{i,j} \sqrt{\sum_{k=1}^{K} | \theta^{k_1} - \theta^{k_2} |} \]

s.t. \( \Theta^{k_c} \in \mathbb{S}_{++}^{n_c}, \) for \( k = 1, \ldots, K \) and \( c = 1, 2. \)

(19)

Given the sample covariance matrices \( \left\{ S^{k_c} \right\}_{k=1}^{K} \) and the two parameters \( \lambda_1 \) and \( \lambda_2, \) we can find the estimates of precision matrices \( \left\{ \Theta^{k_c} \right\}_{c=1}^{2} \) and their sparse approximation \( \left\{ Z^{k_c} \right\}_{c=1}^{2} \) using Algorithm 2. Here we first decomposed (19) into \( K \) individual subproblems (4) using the method of local linear approximation (see the main text). Then we
iteratively solve each subproblems until convergence. Because the estimates \( \hat{\Theta}^{k\circ} \) should be not accurately sparse when they are obtained by using the ADMM algorithm, we use \( \{ \hat{Z}^{k\circ} \}_{k=1,...,K} \) in the experiments, which are introduced as sparse approximation of \( \{ \hat{\Theta}^{k\circ} \}_{k=1,...,K} \) by the ADMM algorithm (please refer to Algorithm 1).

Algorithm 2 Complete algorithm of the TDJGL model (19)

- **Inputs:** The \( 2K \) sample covariance matrices \( \{ S^{k\circ} \}_{k=1,...,K} \) and the regularization parameters \( \lambda_1 \) and \( \lambda_2 \)
- **Output:** The \( 2K \) estimated precision covariance matrices \( \{ \hat{\Theta}^{k\circ} \}_{k=1,...,K} \) and their sparse approximation \( \{ \hat{Z}^{k\circ} \}_{k=1,...,K} \)
- **Main algorithms:**
  1. Initialize \( \hat{\Theta}^{k\circ} \) for \( k=1,\ldots,K \) and \( c=1,2 \).
  2. Compute the weights \( \omega_{ij} = \frac{1}{2\sqrt{\sum_{c=1}^{2} \theta_{ij}^{c}}} \) and \( \psi_{ij} = \frac{1}{2\sqrt{\sum_{c=1}^{2} \theta_{ij}^{c} - \theta_{ij}^{c}}} \) for \( i,j = 1,\ldots,p \).
  3. Update \( \hat{\Theta}^{k1} \) and \( \hat{\Theta}^{k2} \) for all \( k=1,\ldots,K \) by solving problem (4) using Algorithm 1.
  4. Repeat Steps 2 and 3 until the convergence condition is achieved.
  5. Output \( \{ \hat{\Theta}^{k\circ} \}_{k=1,...,K} \) and \( \{ \hat{Z}^{k\circ} \}_{k=1,...,K} \)

2.4 Model selection

For the TDJGL model, parameter \( \lambda_1 \) controls the sparsity of the estimated networks, and parameter \( \lambda_2 \) has an influence on the sparsity of resulting differential networks. Therefore, the choice of \( \lambda_1 \) and \( \lambda_2 \) is critical. We determine the values of parameters in data-driven method via stability selection [6]. Stability selection, which seeks the parameters leading to most stable set of edges, has better results for network inference than other model selection methods including cross validation, Aikake information criterion and Bayesian information criterion [7, 8, 9, 10].

We choose \( \lambda_1 \) and \( \lambda_2 \) so as to use the least amount of regularization that simultaneously makes network sparse and stable. Here we resort to a recently developed stability selection method called StARS [7]. Because \( \lambda_1 \) mainly influences the sparsity and stability of resulting gene networks, while \( \lambda_2 \) mainly controls the sparsity and stability of estimated differential networks, we determine their values separately. We first determine the value of \( \lambda_1 \) while setting \( \lambda_2 = 0 \). Then we determine the value of \( \lambda_2 \) while setting \( \lambda_1 \) with the value chosen in the previous step.

We draw \( S \) random sample sets \( D_1,\ldots,D_S \) from the \( n = n_1 + n_2 \) patients, each of size \( 0.8n \). For now, we choose \( \lambda_1 \) from a given vector of regularization parameter \( \Lambda_1 \) with setting \( \lambda_2 = 0 \). We estimate \( 2K \) networks \( \{ \hat{E}^{k\circ}_{s} (\lambda_1, \lambda_2) \}_{k=1,...,K} \) for each \( D_s \) and each \( \lambda_1 \) from \( \Lambda_1 \). The optimal value of \( \lambda_1 \) controls the average variance over the edges of the networks inferred from sub-sampled data:

\[
\lambda_{1,\text{opt}}^{(\lambda_2)} = \arg \min_{\gamma \in \Lambda_1} \left\{ \max_{\lambda_2 \geq 1} \left( \sum_{k=1}^{K} \sum_{i<j}^{2} \left( \sum_{i<j}^{2} 2\hat{\alpha}_{ij}^{k\circ} (\lambda_1, \lambda_2) \left( 1 - \hat{\alpha}_{ij}^{k\circ} (\lambda_1, \lambda_2) \right) \right) / \binom{p}{2} / 2K \right) \leq \beta \right\},
\tag{20}
\]

where \( \hat{\alpha}_{ij}^{k\circ} (\lambda_1, \lambda_2) = \frac{1}{S} \sum_{s=1}^{S} I \left( (i,j) \in \hat{E}^{k\circ}_{s} (\lambda_1, \lambda_2) \right) \). Here we present StARS using a completeness form. Interested reader is referred to [7].

After determining \( \lambda_1 \), we choose \( \lambda_2 \) from a given vector of regularization parameter \( \Lambda_2 \) according to stability of inferred differential networks. For now, we set \( \lambda_1 = \lambda_{1,\text{opt}}^{(\lambda_2)} \) which is determined above. We estimate \( 2K \) networks \( \{ \hat{E}^{k\circ}_{s} (\lambda_1, \lambda_2) \}_{k=1,...,K} \) for each \( D_s \) and each \( \lambda_2 \) from \( \Lambda_2 \). Then we construct \( K \) differential networks, \( \{ \hat{\mathbf{DE}}^{k\circ}_{s} (\lambda_1, \lambda_2) \}_{k=1,...,K} \), based on the estimated networks. The optimal value for \( \lambda_2 \) is chosen according to the average variance over the differential edges inferred from sub-sampled data:

\[
\lambda_{2,\text{opt}}^{(\lambda_1)} = \arg \min_{\gamma \in \Lambda_2} \left\{ \max_{\lambda_2 \geq 1} \left( \sum_{k=1}^{K} \left( \sum_{i<j}^{2} 2\hat{\beta}_{ij}^{k\circ} (\lambda_1, \lambda_2) \left( 1 - \hat{\beta}_{ij}^{k\circ} (\lambda_1, \lambda_2) \right) \right) / \binom{p}{2} / K \right) \leq \beta \right\},
\tag{21}
\]

where \( \hat{\beta}_{ij}^{k\circ} (\lambda_1, \lambda_2) = \frac{1}{S} \sum_{s=1}^{S} I \left( (i,j) \in \hat{\mathbf{DE}}^{k\circ}_{s} (\lambda_1, \lambda_2) \right) \).
In this study, we set the number of random sample sets $N = 20$ and the stability parameter $\beta = 0.01$.

2.5 Criteria for platinum response groups

The criterion that has been used in [11, 12] is adopted to define platinum-based chemotherapy response groups. In particular, we download the clinical information (Biotab format) of ovarian tumors from the TCGA website. We obtain the drug information from the ‘'nationwidechildrens.org_clinical_drug_ov.txt'' file. Here we only consider platinum-based drugs (carboplatin, cisplatin, carbo) with regimen indication ‘'ADJUVANT’' and ‘'OTHER, SPECIFY IN NOTES’'. The cancer progression information are obtained from the ‘'nationwidechildrens.org_clinical_follow_up_v1.0_nre_ov.txt'' file. New tumors with “new_tumor_event_dx_evidence = [Not Available]” and “new_neoplasms_event_type = [Unknown]” are not considered. The follow-up information of tumors with no progression are obtained from the ‘‘nationwidechildrens.org_clinical_follow_up_v1.0_ov.txt’’ file. For tumors with progression, it is defined as platinum-resistant if the new tumor occurs within 6 months of the end of primary treatment ($days_{to\_new\_tumor\_event\_after\_initial\_treatment} – days_{to\_drug\_therapy\_end} \leq 180 \text{ days}$), and it is defined as platinum-sensitive otherwise ($days_{to\_new\_tumor\_event\_after\_initial\_treatment} – days_{to\_drug\_therapy\_end} > 180 \text{ days}$). For tumors with no progression, it is defined as platinum-sensitive if the follow-up interval is at least 6 months from the date of last primary treatment ($days_{to\_last\_followup} – days_{to\_drug\_therapy\_end} > 180 \text{ days}$). Among the 514 tumors that have all the three types of gene expression profiles, 340 tumors that have explicit cisplatin status, with 242 platinum-sensitive tumors and 98 platinum-resistant tumors. The sensitive and resistant information for each sample is presented in Supplementary information 2.

2.6 Comparison with other graphical lasso models on the ovarian cancer data

We compare TDJGL with FGL, GGL and GL on the ovarian cancer data. For FGL, we run it separately for each platform and each time it is applied across all the three platforms. For GL, we run it separately for each patient group and each platform type. In order to provide interpretable results, we select the tuning parameters of the compared methods to give the similar number of edges and differential edges as those of TDJGL. Unlike TDJGL and FGL, GGL and GL cannot control the similarities of precision matrices between different patient groups. They tend to identify too much differential edges. To better interpret the results of GGL and GL, we sort the absolute values of differential scores in decreasing order, and take the top #DE (the number of differential edges identified by TDJGL) edges for each model.

A common challenge in evaluating gene network inference and differential network analysis using real data is the lack of the gold standards. That is, in our ovarian cancer data analysis, we cannot obtain the true gene networks in the platinum-resistant tumors and the platinum-sensitive tumors. Therefore, it is difficult to compare different methods in terms of the accuracy of identifying group-specific gene networks and differential networks. In this study, we adopt an alternative way to evaluate performance. First, we compare the methods based on the overlaps between edges inferred from different platforms, which can assess the consistency. A method that produces a greater number of edges shared by different platforms is more consistent. Then, we compare the hub nodes in the differential networks in terms of known drug resistance-related genes and cancer-related genes. A method that works better in capturing known functionally important genes in the differential networks might have better performance in inferring the differential networks.

We observe that overlaps between edges (and differential edges) identified by FGL and GL from the different platforms are quite low (Figures S7 and S9). For GGL which encourages a similar network structure across all platforms, more than half of identified edges are shared by all the three platforms (Figure S8 (a)-(b)). Because GGL does not consider the similarity of differential networks across platforms, a great number of differential edges detected by GGL are supported by only one platform (Figure S8 (c)). As mentioned in the main text, both gene networks and differential networks inferred by TDJGL share a great number of edges across all the three platforms (Figure S6). We also compare the hub nodes in the differential networks inferred by different methods. For FGL, GGL and GL, we consider the 18 genes which have the largest degree of connectivity as hub genes. From Table S1, we find that the set of hub genes determined by TDJGL includes more cisplatin resistance-related genes, drug resistance-related genes and cancer-related genes than those determined by the other three methods.

Table S1: The number of hub genes that have been reported as platinum resistance-related genes, drug resistance-related genes and cancer-related genes

| Methods | GEAR_{cisplatin} | GEAR_{drug} | CGC |
|---------|-----------------|-------------|-----|
| TDJGL   | 5               | 10          | 8   |
| FGL     | 0               | 3           | 6   |
| GGL     | 1               | 5           | 6   |
| GL      | 1               | 3           | 4   |
Figure S7: Overlaps between the edges (and differential edges) detected by FGL from the three platforms for (a) platinum-resistant tumors, (b) platinum-sensitive tumors and (c) differential networks.

Figure S8: Overlaps between the edges (and differential edges) detected by GGL from the three platforms for (a) platinum-resistant tumors, (b) platinum-sensitive tumors and (c) differential networks.

Figure S9: Overlaps between the edges (and differential edges) detected by GL from the three platforms for (a) platinum resistant tumors, (b) platinum-sensitive tumors and (c) differential networks.

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] Karthik Mohan, Palma London, Maryam Fazel, Daniela Witten, and Su-In Lee. Node-based learning of multiple gaussian graphical models. The Journal of Machine Learning Research, 15(1):445–488, 2014.

[3] 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.
[4] Daniela M Witten and Robert Tibshirani. Covariance-regularized regression and classification for high dimensional problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):615–636, 2009.

[5] Holger Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.

[6] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[7] Han Liu, Kathryn Roeder, and Larry Wasserman. Stability approach to regularization selection (stars) for high dimensional graphical models. In *Advances in neural information processing systems*, pages 1432–1440, 2010.

[8] Genevera I Allen and Zhandong Liu. A local poisson graphical model for inferring networks from sequencing data. *IEEE Transactions on NanoBioscience*, 12(3):189–198, 2013.

[9] Marinka Žitnik and Blaž Zupan. Gene network inference by fusing data from diverse distributions. *Bioinformatics*, 31(12):i230–i239, 2015.

[10] Zachary D Kurtz, Christian L Müller, Emily R Miraldi, Dan R Littman, Martin J Blaser, and Richard A Bonneau. Sparse and compositionally robust inference of microbial ecological networks. *PLOS Computational Biology*, 11(5):e1004226, 2015.

[11] The Cancer Genome Atlas Research Network. Integrated genomic analyses of ovarian carcinoma. *Nature*, 474(7353):609–615, 2011.

[12] Sheida Nabavi, Daniel Schmolze, Mayinuer Maitituoheti, Sadhika Malladi, and Andrew H Beck. Emdomics: a robust and powerful method for the identification of genes differentially expressed between heterogeneous classes. *Bioinformatics*, page btv634, 2015.