Supplemental Material for "Estimating Multiple Precision Matrices with Cluster Fusion Regularization"

Bradley S. Price, Aaron J. Molstad, and Ben Sherwood

1 Proof of GEN-ISTA Convergence

1.1 Proof of Lemma 1

Define

$$\tilde{S}_c = \left\{ S_c - \frac{\lambda_2}{n_c \text{card}(D_q^w)} \left(\sum_{c' \in D_q^w \setminus \{c\}} \Omega_{c'} \right) \right\}, \quad \gamma_{c1} = \frac{\lambda_1}{n_c}, \quad \gamma_{c2} = \frac{\lambda_2(\text{card}(D_q^w) - 1)}{2n_c \text{card}(D_q^w)},$$

so that the argument minimizing the objective function for the block wise coordinate descent step of PCEN can be expressed

$$\underset{\Omega_c \in \mathbb{S}_p^+}{\arg\min} \left\{ \text{tr}(\tilde{S}_c \Omega_c) - \log\det(\Omega_c) + \gamma_{c1} \|\Omega_c\|_1 + \gamma_{c2} \|\Omega_c\|_F^2 \right\}, \tag{1}$$

In this section we show that the optimal solution of (1) is contained on a compact subset of $\mathbb{S}_+^p$. To gain a deeper understanding of the solution we obtain the dual form of (1). Define Z_c to be a symmetric $p \times p$ matrix, then (1) can be rewritten as

$$\min_{\Omega_c \in \mathbb{S}_+^p} \text{tr}(\tilde{S}_c \Omega_c) - \log\det(\Omega_c) + \gamma_{c1} \max_{|\text{vec}(Z_c)|_\infty < 1} \text{tr}(Z_c \Omega_c) + \gamma_{c2} |\Omega_c|_2^2.$$

Just as in Banerjee et al. (2008) we exchange the max and min to obtain the dual problem,

$$\max_{|\text{vec}(Z_c)|_\infty < 1} \min_{\Omega_c \in \mathbb{S}_+^p} \text{tr}\{(\tilde{S}_c + \gamma_{c1} Z_c)\Omega_c\} - \log\det(\Omega_c) + \gamma_{c2} |\Omega_c|_2^2. \tag{2}$$

Notice that the optimization problem in (2) with respect to Ω_c is just a ridge penalized precision matrix estimation problem with tuning parameter γ_{c2}, which was investigated by (Witten and Tibshirani, 2009). Define

$$Q(A, \eta) = \underset{\Theta \in \mathbb{S}_+^p}{\arg\min} \left\{ \text{tr}(A\Theta) - \log\det(\Theta) + \eta |\Theta|_2^2 \right\},$$

Bradley S. Price, Management Information Systems Department, West Virginia University (E-Mail: *brad.price@mail.wvu.edu)*. Aaron J. Molstad, Department of Statistics and Genetics Institute, University of Florida (E-mail: *amolstad@ufl.edu*). Ben Sherwood, School of Business, University of Kansas (E-mail: *ben.sherwood@ku.edu*).

and $\dot{\Omega}_{Z_c} = Q(\tilde{S}_c + \gamma_{c1} Z_c, \gamma_{c2})$, then the dual problem is

$$\max_{|\mathrm{vec}(Z_c)|_\infty < 1} \mathrm{tr}\{(\tilde{S}_c + \gamma_{c1} Z_c)\dot{\Omega}_{Z_c}\} - \log\det(\dot{\Omega}_{Z_c}) + \gamma_{c2}|\dot{\Omega}_{Z_c}|_2^2. \tag{3}$$

Now we are able to show the result.

Proof. Define

$$q(a, \eta) = \frac{-a + \sqrt{a^2 + 8\eta}}{4\eta}.$$

Note that $q(a, \eta) > 0$ for all $a \in \mathbb{R}$ when $\eta > 0$ and given some $b \in \mathbb{R}$ such that $b < a$ then $q(a, \eta) > q(b, \eta) > 0$. Let $\widehat{Z}_c$ be the solution to (3). Then we are able to rewrite $\Omega_c^* = \dot{\Omega}_{\widehat{Z}_c} = Q(\tilde{S}_c + \gamma_{c1}\widehat{Z}_c, \gamma_{c2}) = V\widehat{D}V^T$, where V is a matrix of the eigenvectors of $\tilde{S}_c + \gamma_{c1}\widehat{Z}_c$, and $\widehat{D}$ is a diagonal matrix with jth diagonal equal to $q\left(\rho_j(\tilde{S}_c + \gamma_{c1}\widehat{Z}_c), \gamma_{c2}\right)$ (Witten and Tibshirani, 2009).

To complete the proof all that is left to do is bound the cases of $j = 1$ and $j = p$. Weyl's Theorem provides the inequalities

$$\rho_p(\tilde{S}_c) + \gamma_{c1}\rho_p(\widehat{Z}_c) \leq \rho_p(\tilde{S}_c + \gamma_{c1}\widehat{Z}_c), \tag{4}$$

and

$$\rho_1(\tilde{S}_c + \gamma_{c1}\widehat{Z}_c) \leq \rho_1(\tilde{S}_c) + \gamma_{c1}\rho_1(\widehat{Z}_c). \tag{5}$$

Then, using the inequality

$$-p \leq \rho_p(\widehat{Z}_c) \leq \rho_1(\widehat{Z}_c) \leq p,$$

combined with (4) and (5) yields

$$\begin{aligned} 0 < q(\rho_p(\tilde{S}_c) - \gamma_{c1}p, \gamma_{c2}) &\leq q(\rho_p(\tilde{S}_c) + \gamma_{c1}\rho_p(\widehat{Z}_c), \gamma_{c2}) \\ &\leq q(\rho_1(\tilde{S}_c) + \gamma_{c1}\rho_1(\widehat{Z}_c), \gamma_{c2}) \leq q(\rho_1(\tilde{S}_c) + \gamma_{c1}p, \gamma_{c2}) < \infty. \end{aligned}$$

These resulting bounds are for eigenvalues of Ω^*. Inverting the bounds (in order to simplify the expressions) yields the result of the lemma. □

1.2 Lipschitz Continuity of the $\bigtriangledown f(\Omega)$

In this section, we prove of Lemma 2 from the main mauscript.

Proof. Assume $aI \preceq \Omega_A, \Omega_B \preceq bI$, for some $\Omega_A, \Omega_B \in \mathbb{S}_+^p$ and $a > 0$ and $b < \infty$. Applying Lemma 2 of Rolfs et al. (2012), we have that

$$\frac{1}{b^2}\|\Omega_A - \Omega_B\|_2 \leq \|\Omega_A^{-1} - \Omega_B^{-1}\|_2 \leq \frac{1}{a^2}\|\Omega_A - \Omega_B\|_2.$$

Thus, we have

$$\begin{aligned}
\| \bigtriangledown f(\Omega_A) - \bigtriangledown f(\Omega_B)\|_F &= \|\Omega_B^{-1} - \Omega_A^{-1} + 2\gamma_{c2}(\Omega_A - \Omega_B)\|_F \\
&\leq \sqrt{p}\|\Omega_B^{-1} - \Omega_A^{-1} + 2\gamma_{c2}(\Omega_A - \Omega_B)\|_2 \\
&\leq \sqrt{p}\|\Omega_B^{-1} - \Omega_A^{-1}\|_2 + 2\sqrt{p}\gamma_{c2}\|\Omega_A - \Omega_B\|_2 \\
&\leq \frac{\sqrt{p}}{a^2}\|\Omega_A - \Omega_B\|_2 + 2\sqrt{p}\gamma_{c2}\|\Omega_A - \Omega_B\|_2 \\
&\leq \left(\frac{\sqrt{p}}{a^2} + 2\sqrt{p}\gamma_{c2}\right)\|\Omega_A - \Omega_B\|_F.
\end{aligned}$$

□

1.3 Proof of Theorem 1

First, we will provide a general result that we use to establish the linear convergence rate of our algorithm.

Lemma 4. *Assume that iterates of the algorithm proposed satisfy* $aI \preceq \Omega_c^{(k)} \preceq bI$ *for all* k *and some fixed constants* $0 < a < b < \infty$. *If* $t \leq \frac{a^2}{2\alpha^2\gamma_{c2}+1}$ *then:*

1. $$\|\Omega_c^{(k+1)} - \Omega_c^*\|_F \leq \max\left\{\left|m_t - \frac{t}{a^2}\right|, \left|m_t - \frac{t}{b^2}\right|\right\}\|\Omega_c^{(k)} - \Omega_c^*\|_F,$$

   where $m_t = 1 - 2t\gamma_{c2}$.

2. *The step size* t *that will lead to the optimal worst-case bound is* $t_w = \frac{2}{4\gamma_{c2}+b^{-2}+a^{-2}}$.

3. *The optimal worst case bound is*

$$1 - \frac{2}{1 + \frac{2\gamma_{c2}+a^{-2}}{2\gamma_{c2}+b^{-2}}} < 1.$$

We present the full proof in Section 1.3.1, but first, we make a number of important remakrs. First, we note that if $\gamma_{c2} = 0$, then our result is equivalent to the bounds of Rolfs et al. (2012). Second, we point out that as $\gamma_{c2} \to \infty$, the optimal worst case bounds approach 0. Finally, as γ_{c2} gets larger the maximum step size that is applicable also approaches 0.

1.3.1 Proof of Lemma 4

Our proof strategy is similar to that of Rolfs et al. (2012) but there are differences due to the ridge penalty.

Proof. Recall that $\Omega_c^* = \mathcal{S}\{\Omega_c^* - t(\tilde{S}_c - (\Omega_c^*)^{-1} + 2\gamma\Omega_c^*), t\gamma_{c1}\}$. Let $\Sigma_c^* = [\Omega_c^*]^{-1}$ and $\Sigma_c^{(k)} = [\Omega_c^{(k)}]^{-1}$. By the definitions of $\Omega_c^{(k+1)}$ and $\Omega_c^{(k+\frac{1}{2})}$ and Lemma 2.2 from Combettes and Wajs

(2005)

$$\begin{aligned}
\|\Omega_c^{(k+1)} - \Omega_c^*\|_F &= \|\mathcal{S}(\Omega_c^{(k+\frac{1}{2})}, t\gamma_{c1}) - \mathcal{S}\{\Omega_c^* - t(\tilde{S}_c - \Sigma_c^* + 2\gamma\Omega_c^*), t\gamma_{c1}\}\|_F \\
&\leq \|\Omega_c^{(k+\frac{1}{2})} - \{\Omega_c^* - t(\tilde{S}_c - \Sigma_c^* + 2\gamma\Omega_c^*)\}\|_F \\
&= \|\Omega_c^{(k+\frac{1}{2})} + t\tilde{S}_c - \{(1-2t\gamma)\Omega_c^* + t\Sigma_c^*\}\|_F \\
&= \|\Omega_c^k - t\{\tilde{S}_c - \Sigma_c^{(k)} + 2\gamma_{c2}\Omega_c^{(k)}\} + t\tilde{S}_c - \{(1-2t\gamma)\Omega_c^* + t\Sigma_c^*\}\|_F \\
&= \|\{\Omega_c^{(k)} - t\left(2\gamma_{c2}\Omega_c^{(k)} - \Sigma_c^{(k)}\right)\} - \{\Omega_c^* - t\left(2\gamma_{c2}\Omega_c^* - \Sigma_c^*\right)\}\|_F \\
&= \|\{(1-2t\gamma_{c2})\Omega_c^{(k)} + t\Sigma_c^{(k)}\} - \{(1-2t\gamma_{c2})\Omega_c^* + t\Sigma_c^*\}\|_F.
\end{aligned}$$

If $h : U \subset \mathbb{R}^{p^2} \to \mathbb{R}^m$ is a differentiable mapping with Jacobian J_h, $x, y \in U$ and $vx + (1-v)y \in U$ for all $v \in [0,1]$, then

$$\|h(x) - h(y)\|_2 \leq \sup_{v\in[0,1]} \{\|J_h(vx + (1-v)y)\|_2\}\|x - y\|_2.$$

Recall $m_t = 1 - 2t\gamma_{c2}$ and define

$$h_{\gamma_{c1},\gamma_{c2}}\{\mathrm{vec}(\Omega_c)\} = m_t\,\mathrm{vec}(\Omega_c) + t\,\mathrm{vec}(\Omega_c^{-1}).$$

Note that,

$$J_{h_{\gamma_{c1},\gamma_{c2}}}(\Omega_c) = m_t I_{p^2} - t\Omega_c^{-1} \otimes \Omega_c^{-1}.$$

For $v \in [0,1]$ let

$$H_{k,v} = \mathrm{vec}\{v\Omega_c^{(k)} + (1-v)\Omega_c^*\},$$

it follows that

$$\|h_{\gamma_{c1},\gamma_{c2}}(\Omega_c^{(k)}) - h_{\gamma_{c1},\gamma_{c2}}(\Omega_c^*)\|_2 \leq \sup_{v\in[0,1]} \{\|m_t I_{p^2} - tH_{v,k}^{-1} \otimes H_{v,k}^{-1}\|_2\}\|\Omega_c^{(k)} - \Omega_c^*\|_F.$$

Therefore, for any value of k and v

$$\min\{\rho_p(\Omega_c^{(k)}), \rho_p(\Omega_c^*)\} \leq \rho_p(H_{k,v}) \leq \rho_1(H_{k,v}) \leq \max\{\rho_1(\Omega_c^{(k)}), \rho_1(\Omega_c^*)\}.$$

Combining these results, we obtain

$$\sup_{v\in[0,1]} \{\|m_t I_{p^2} - tH^{-1} \otimes H^{-1}\|_2\} \leq \max\left\{\left|m_t - \frac{t}{b^2}\right|, \left|m_t - \frac{t}{a^2}\right|\right\},$$

which proves part 1 of Lemma 4. We can further show that the algorithm converges linearly if

$$s(t) = \max\left\{\left|m_t - \frac{t}{b^2}\right|, \left|m_t - \frac{t}{a^2}\right|\right\} \in (0,1),\ \forall k.$$

The minimum of $s(t)$ is obtained at

$$t_w = \frac{2}{4\gamma_{c2} + b^{-2} + a^{-2}},$$

and then evaluating $s(t_w)$ completes the result.

□

1.3.2 Proof of Lemma 3

In this section we assume that the eigenvalues of $\Omega_c^{(k)}$ are bounded for all k and recall that, letting $\Sigma_c^{(k)} \equiv (\Omega_c^{(k)})^{-1}$,

$$\Omega_c^{(k+\frac{1}{2})} = \Omega_c^{(k)} - t(\tilde{S}_c - \Sigma_c^{(k)} + 2\gamma_{c2}\Omega_c^{(k)}). \tag{6}$$

Lemma 5. *Assume $0 < a < b < \infty$ are known such that $aI \preceq \Omega_c^{(k)} \preceq bI$, and that $t > 0$. Then the eigenvalues of $\Omega_c^{(k+\frac{1}{2})}$, which is defined by (6), satisfy:*

$$\rho_p(\Omega_c^{(k+\frac{1}{2})}) \geq \begin{cases} \sqrt{\frac{t}{1-2t\gamma_{c2}}} - \frac{t}{\sqrt{\frac{t}{1-2t\gamma_{c2}}}} - t\rho_1(\tilde{S}_c), & \text{if } a \leq \sqrt{\frac{t}{1-2t\gamma_{c2}}} \leq b \\ \min\left\{m_t a + \frac{t}{a}, m_t b + \frac{t}{b}\right\} - t\rho_1(\tilde{S}_c) & \text{otherwise} \end{cases}$$

and

$$\rho_1(\Omega_c^{(k+\frac{1}{2})}) \leq \max\left\{m_t a + \frac{t}{a}, m_t b + \frac{t}{b}\right\} - t\rho_p(\tilde{S}_c).$$

Proof. Define the spectral decomposition of $\Omega_c^{(k)} = UDU^T$, then

$$\begin{aligned} \Omega_c^{(k+\frac{1}{2})} &= \Omega_c^{(k)} - t(\tilde{S}_c - \Sigma_c^{(k)} + 2\gamma_{c2}\Omega_c^{(k)}) \\ &= UDU^T - t(\tilde{S}_c - UD^{-1}U^T + 2\gamma_{c2}UDU^T) \\ &= U\{D - t(U^T\tilde{S}_cU - D^{-1} + 2\gamma_{c2}D)\}U^T. \end{aligned}$$

Next, by Wyel's Theorem it follows that

$$\rho_p(\Omega_c^{(k+\frac{1}{2})}) \geq \rho_p(\Omega_c^{(k)}) - t\left\{2\gamma_{c2}\rho_p(\Omega_c^{(k)}) - \frac{1}{\rho_p(\Omega_c^{(k)})} + \rho_1(\tilde{S}_c)\right\},$$

and

$$\rho_1(\Omega_c^{(k+\frac{1}{2})}) \leq \rho_1(\Omega_c^{(k)}) - t\left\{2\gamma_{c2}\rho_1(\Omega_c^{(k)}) - \frac{1}{\rho_1(\Omega_c^{(k)})} + \rho_p(\tilde{S}_c)\right\}.$$

Recalling that $m_t = 1 - 2t\gamma_{c2}$, the function

$$r(x) = m_t x + \frac{t}{x}, \;\; a \leq x \leq b,$$

has a global minimum at $x_w = \sqrt{\frac{t}{1-2t\gamma_{c2}}}$. Thus, using arguments similar to Rolfs et al. (2012) proof of Lemma 4, we have that

$$\rho_p(\Omega_c^{(k+\frac{1}{2})}) = \begin{cases} \sqrt{\frac{t}{1-2t\gamma_{c2}}} - \frac{t}{\sqrt{\frac{t}{1-2t\gamma_{c2}}}} - t\rho_1(\tilde{S}_c), & \text{if } a \leq \sqrt{\frac{t}{1-2t\gamma_{c2}}} \leq b \\ \min\left\{m_t a + \frac{t}{a}, m_t b + \frac{t}{b}\right\} - t\rho_1(\tilde{S}_c) & \text{otherwise} \end{cases}$$

and

$$\rho_1(\Omega_c^{(k+\frac{1}{2})}) \leq \max\left\{m_t a + \frac{t}{a}, m_t b + \frac{t}{b}\right\} - t\rho_p(\tilde{S}_c),$$

which obtain the bounds. □

Next we need to show that when the full step is taken by soft thresholding the eigenvalues are bounded.

Lemma 6. *Assume $0 < a < b$ and $t, m_t = 1 - 2t\gamma_{c2} > 0$ then $\min\left\{m_t a + \frac{t}{a}, m_t b + \frac{t}{b}\right\} = m_t a + \frac{t}{a}$ if and only if $t \leq \frac{ab}{1+2\gamma_{c2}ab}$*

Proof. Using the assumptions we have that

$$\begin{aligned} m_t a + \frac{t}{a} \leq m_t b + \frac{t}{b} \quad &\iff \quad t\left(\frac{1}{a} - \frac{1}{b}\right) \leq m_t(a-b) \\ &\iff \quad t \leq m_t ab \\ &\iff \quad t \leq \frac{ab}{1+2\gamma_{c2}ab}. \end{aligned}$$

from which the result follows. □

Next, for the sake of convenience, we restate Lemma 6, a useful result on soft-thresholding, from the Supplementary Material of Rolfs et al. (2012).

Lemma 7. *Let A be a symmetric $p \times p$ matrix and $\delta > 0$. Then the soft-thresholded matrix $\mathcal{S}(A, \delta)$ satisfies $\rho_p(A) - \delta p \leq \rho_p\{\mathcal{S}(A, \delta)\}$. Moreover, the soft-thresholded matrix is positive definite if $\rho_p(A) > \delta p$ (Rolfs et al., 2012).*

Proof. Proof is in the supplementary material of Rolfs et al. (2012). □

Lemma 8. *Let $\gamma_{c1} > 0$ and α be the same as defined in Lemma 1. Assume $\alpha < b'$ and $\alpha I \preceq \Omega_c^{(k)} \preceq b' I$ and recall that*

$$\Omega_c^{(k+1)} = \mathcal{S}(\Omega_c^{(k+\frac{1}{2})}, t\gamma_{c1}),$$

where $\Omega_c^{(k+\frac{1}{2})}$ is defined by (6). Then for every $0 < t \leq \frac{\alpha^2}{2+\gamma_{c2}\alpha^2+1}$, then $\alpha I \preceq \Omega_c^{(k+1)}$.

Proof. Lemma 6 gives us

$$\min\left\{m_t\alpha + \frac{t}{\alpha}, m_t b + \frac{t}{b'}\right\} = m_t\alpha + \frac{t}{\alpha}.$$

since $t \leq \frac{\alpha^2}{2\gamma_{c2}\alpha^2+1} \leq \frac{\alpha b'}{2\gamma_{c2}\alpha b'+1}$. Note that $0 < t \leq \frac{\alpha^2}{2\gamma_{c2}\alpha^2+1}$ guarantees that $\sqrt{\frac{t}{1-2t\gamma_{c2}}} \leq \alpha$. Therefore by Lemma 5

$$\rho_p(\Omega_c^{(k+\frac{1}{2})}) \geq m_t\alpha + \frac{t}{\alpha} - t\rho_1(\tilde{S}_c).$$

We continue by applying Lemma 7 to $\Omega_c^{(k+1)}$ where we obtain

$$\begin{aligned}\rho_p(\Omega_c^{(k+1)}) &\geq \rho_p(\Omega_c^{(k+\frac{1}{2})}) - p\gamma_{c1}t \\ &\geq m_t\alpha + \frac{t}{\alpha} - t\rho_1(\tilde{S}_c) - p\gamma_{c1}t.\end{aligned}$$

Therefore, we have that $\alpha I \preceq \Omega_c^{(k+1)}$ when

$$m_t\alpha + \frac{t}{\alpha} - t\rho_1(\tilde{S}_c) - p\gamma_{c1}t \geq \alpha,$$

or equivalently

$$-2\gamma_{c2}t\alpha + \frac{t}{\alpha} - t\rho_1(\tilde{S}_c) - p\gamma_{c1}t \geq 0.$$

Since $t > 0$ we may reorganize this a final time as

$$-2\gamma_{c2}\alpha + \frac{1}{\alpha} - \left(\rho_1(\tilde{S}_c) + p\gamma_{c1}\right) \geq 0.$$

Solving for α we have that $\alpha I \preceq \Omega_c^{(k+1)}$ if

$$\alpha \leq \frac{1}{q(\rho_1(\tilde{S}_c) + \gamma_{c1}p, 2\gamma_{c2})},$$

which holds by Lemma 1. □

Lemma 9. *Let α be the same as in Lemma 1 and $t \leq \frac{\alpha^2}{2+\gamma_{c2}\alpha^2+1}$. The the proposed algorithm iterates satisfy $\Omega_c^{(k)} \preceq b'I$ for all k where $b' = \|\Omega_c^*\|_2 + \|\Omega_c^{(0)} - \Omega^*\|_F$.*

Proof. Using results from Lemma 1 and Lemma 8 we have that

$$\Lambda_k^+ = \max\{\rho_1(\Omega_c^{(k)}), \rho_1(\Omega_c^*)\} > \Lambda_k^- = \min\{\rho_p(\Omega_c^{(k)}), \rho_p(\Omega_c^*)\} \geq \alpha^2.$$

Since $t \leq \frac{\alpha^2}{2+\gamma_{c2}\alpha^2+1}$,

$$\max\left\{\left|1-\frac{t}{b^2}\right|, \left|1-\frac{t}{a^2}\right|\right\} \leq 1.$$

Next, by applying Theorem 3

$$\|\Omega_c^{(k)} - \Omega_c^*\|_F \leq \|\Omega_c^{(k-1)} - \Omega_c^{(k)}\|_F.$$

Finally we have that,

$$\|\Omega_c^{(k)}\|_2 - \|\Omega_c^*\|_2 \leq \|\Omega_c^{(k)} - \Omega_c^*\|_2 \leq \|\Omega_c^{(0)} - \Omega_c^*\|_F,$$

from which we obtain the bound

$$\rho_1(\Omega_c^{(k)}) \leq \|\Omega_c^*\|_2 + \|\Omega_c^{(0)} - \Omega_c^*\|_F.$$

□

Finally we will formally prove Lemma 3.

Proof. Applying the results of Lemma 8 and Lemma 9 we have that

$$\alpha I \preceq \Omega_c^{(k)} \preceq b' I,$$

and

$$b' \leq \|\Omega_c^*\|_2 + \sqrt{p}\|\Omega_c^{(0)} - \Omega_c^*\|_2 \leq \beta + \sqrt{p}(\beta - \alpha).$$

□

2 GEN-ISTA Algorithm with Backtracking

We now state the GEN-ISTA algorithm with backtracking line search. Recall that $\Sigma_c^{(k)} \equiv (\Omega_c^{(k)})^{-1}$.

1. Initialize, $k = 0$, $\eta \in (0, 1)$, $\epsilon > 0$, $t_0 > 0$, and $\Omega_c^{(0)} \in \mathbb{S}_+^p$.
2. While $|f(\Omega_c^{(k)}) - f(\Omega_c^{(k+1)})| > \epsilon$ or $k < 1$
   (a) Set $t = t_0$
   (b) $\Omega_c^{(k+\frac{1}{2})} = \Omega_c^{(k)} - t\{\tilde{S}_c - \Sigma_c^{(k)} + 2\gamma_{c2}\Omega_c^{(k)}\}$
   (c) $\Omega_c^{(k+1)} = \mathcal{S}(\Omega_c^{(k+\frac{1}{2})}, t\gamma_{c1})$
   (d) If $\Omega_c^{(k+1)} \notin \mathbb{S}_+^p$, then update $t = t\eta$ and return to Step 2 (b). Else, continue to Step 2 (e)

(e) If $f(\Omega_c^{(k+1)}) - f(\Omega_c^{(k)}) > + \text{tr}\{(\Omega_c^{(k+1)} - \Omega_c^{(k)})^T(\tilde{S}_c - \Sigma_c^{(k)} + 2\gamma\Omega_c^{(k)})\} + \frac{1}{2t}\|\Omega_c^{(k+1)} - \Omega_c^{(k)}\|_F$, then update $t = t\eta$ and return to Step 2 (b). Else, continue to Step 2 (f).

(f) If not converged, update $k = k + 1$ and return to Step 2 (a).

3 Additional simulation results

3.1 Gaussian graphical model simulation: Two clusters, block diagonal structures

In the final setting, we again assume a data generating model where the four precision matrices are divided into two groups. We generate Ω_{*1} such that it is block diagonal with each block size of $p/2 \times p/2$. The first block is generated using $U = E(A_1, p/2)$, and the second block is the identity matrix, where A_1 is an adjacency matrix from an Erdos Renyi, with $p/2$ connections. Using Ω_{*1} we generate Ω_{*2} such that it is block diagonal with block size $p/2 \times p/2$. We define the upper block of Ω_{*2} as $R\left(A_3, L, (-.01, .01)\right)$, and the lower block to be the identity where A_3 is the adjacency matrix A_1 with four edges removed. Next, Ω_{*3} is generated in a similar way to Ω_{*1} and Ω_{*4} is generated from Ω_{*3} in the same fashion Ω_{*2} is generated from Ω_{*1}.

The results in panels (e) and (f) Figure 1 are average log sum of squared Frobenius norm error and the average true positive rate as the number of non-zero elements in the precision matrices varying with $p = 100$ and $n = 200$. The results for the case of $p = 20$ and $p = 50$ can be found in the Supplementary Material. Results exhibit a similar pattern to the results displayed in Sections 4.2-4.4 For certain values of λ_2, PCEN-2 is competitive in estimation and graph recovery with the other methods, specifically LASICH-OR. As p increases, we see the estimation and graph recovery of PCEN decreases relative to LASICH-OR, but is still competitive with other competitors. Again, this can be attributed to LASICH-OR having oracle information and its use of the group penalty which exploits similar sparsity patterns across all precision matrices.

3.2 Timing Results of PCEN

In this section we present results of the timing results from the simulations found in Section 4 of the manuscript. Figures 2-5 present the average time in seconds for convergence for PCEN with $Q = 2$ and PCEN $Q = 3$ for $\lambda_2 \in 10^{\{-3,1,3\}}$ and λ_1 that varies. Figure 2 represents timings associated with the simulation presented in Section 4.2, Figure 3 presents the results associated with the simulation described in Section 4.3, Figure 4 shows the results of the simulation described in Section 3.1 of the supplemental material, and Figure 5 shows the results of the simulation described in Section 4.4 of the manuscript. Results are consistent across all simulation settings and p. As λ_1 gets extremely large the algorithm slows down, this is due to the diagonal penalization. For larger values of λ_2 we see an increase in time as well. This is due to the restriction in similarity. For extremely large values of λ_1 we see a decrease again in speed due to the completely sparse estimates. We note that using different

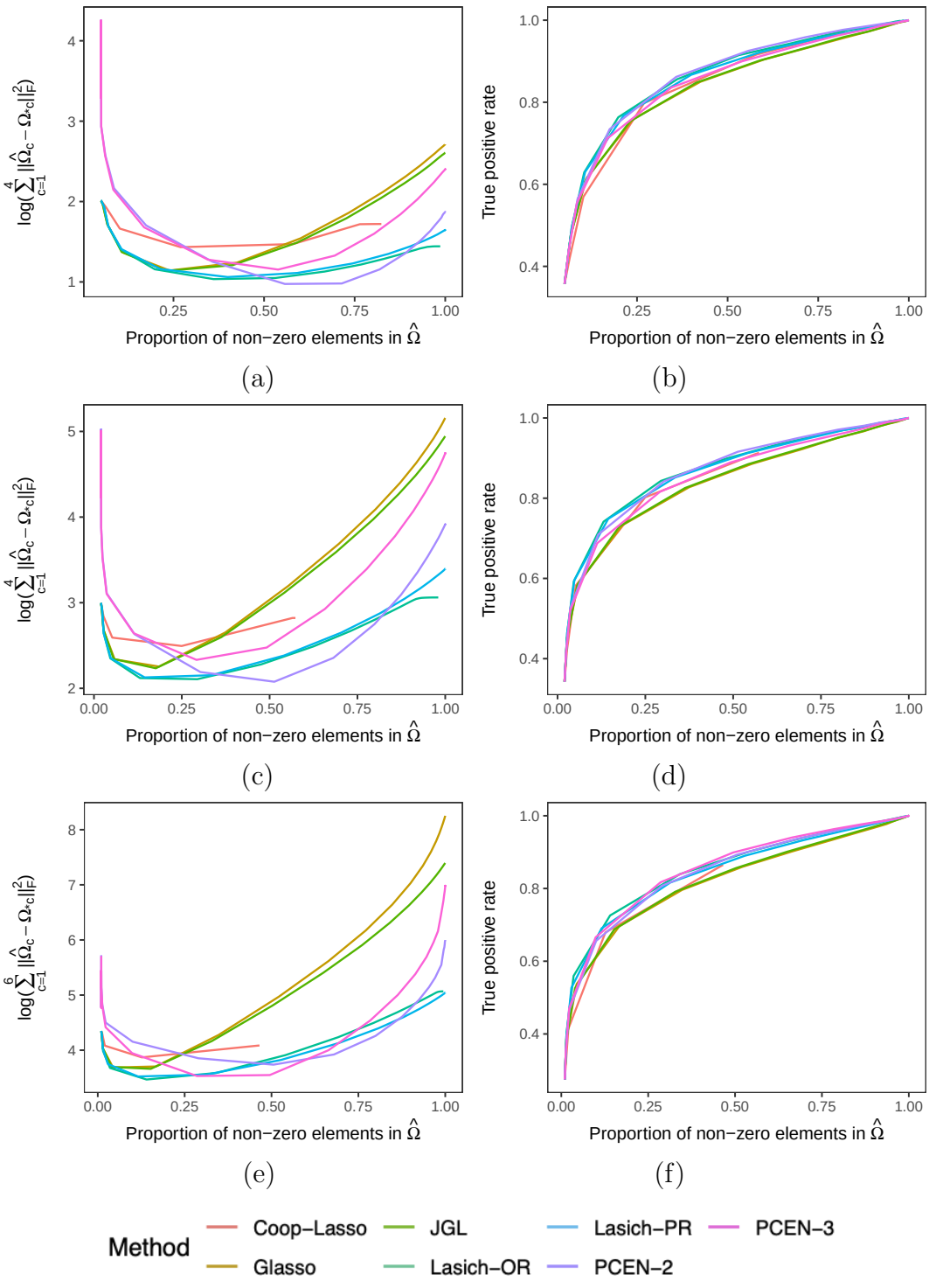


Figure 1: Results for the simulation setting described in Section 3.1 in the Supplement (a,b) consider $p = 20$, (c,d) consider $p = 50$, and (e,f) consider when $p = 100$. Each line represents the average of 50 replications of the denoted method when λ_2 is fixed, and λ_1 varies.

initialization can lead to different speed increases. For instance initialization at an elastic net estimate instead of a ridge estimate can lead to different number of iterations, which will effect speed. In each of these cases converges of our iterative algorithm was reached within 3-5 iterations.

3.3 PCEN Simulation Cluster Detection

This section we present the results on cluster detection from the simulations shown in Section 4 of the manuscript. Figure 6 presents the proportion of the 50 replications that the correct clustering was detected for each p for the correct Q. In the case of panels (a)-(c) this corresponds to $Q = 2$ and for panel (d) this corresponds to $Q = 3$. All methods use $\lambda_2 = 10^3$ which corresponded to the lowest forbenious norm error in each simulation for varying λ_1. The results are consistent across the simulations, for $p = 20$ and 50 the methods are able to detect the correct clustering for the smaller values of λ_1 this corresponds to the more dense estimates of $\hat{\mathbf{\Omega}}$ that we see in the results presented in Section 4. As the estimates become more sparse, the methods are unable to detect the correct clustering. For each setting when $p = 100$ we see a drop off in detection ability for dense estimates of $\hat{\mathbf{\Omega}}$, while increasing to perfect detection at certain values of λ_1 these values correspond to lower estimates of forbenious norm error seen in Section 4. We see a much larger drop off in panel (d) which corresponds to $Q = 3$, $C = 6$. Again we see for extreme levels of sparsity the clustering is missed.

3.4 QDA under clustered, dense, and ill-conditioned precision matrices

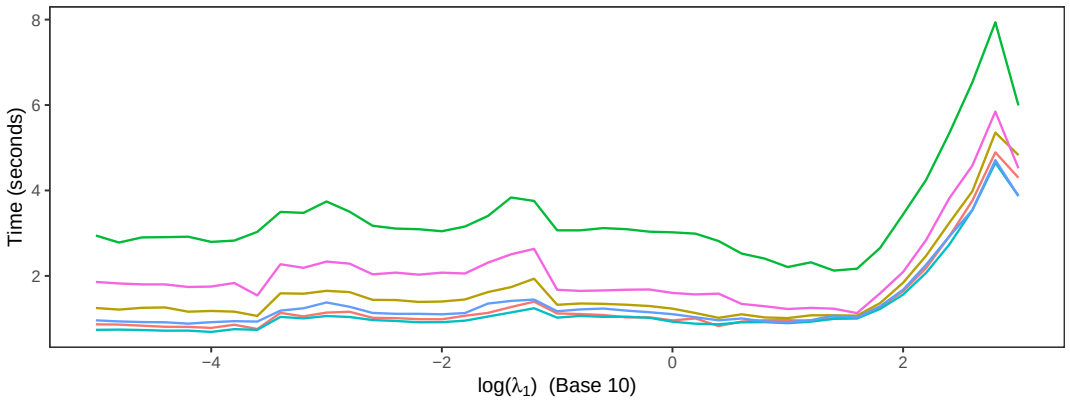
In this section, we present additional simulation results For 100 independent replications, we generate $Z_1, Z_2 \in \mathbb{R}^{100\times p}$ where each row is an independent realization from $\mathrm{N}_p(0, I_p)$. We then obtain V_1 and V_2, the right singular vectors of Z_1, Z_2 respectively. Separately, define the function

$$D(g_1, g_2, j) = g_1\frac{p-j+1}{p}I\{1 \le j \le 6\}+g_2\frac{p-j+1}{p}I\{7 \le j \le 11\}+\frac{p-j+1}{p}I\{12 \le j \le p\}.$$

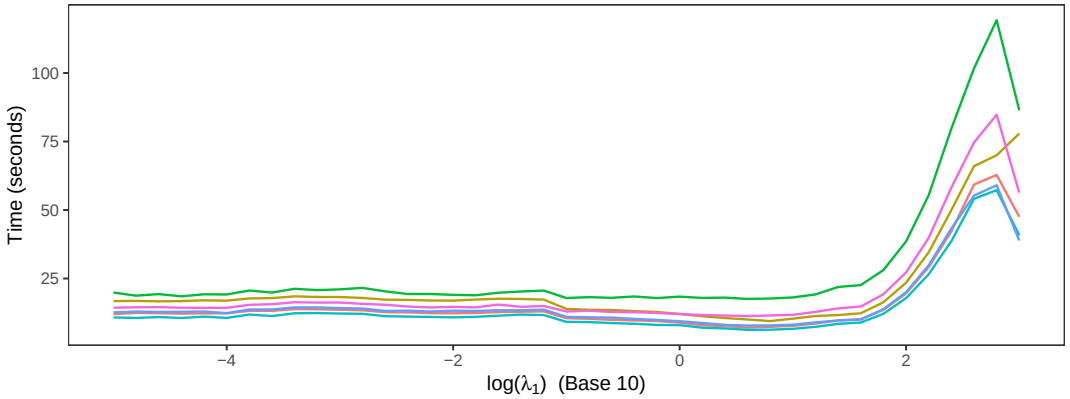
Finally, we set $\Sigma_{*1} = V_1^T H_1 V_1, \Sigma_{*2} = V_1^T H_2 V_1, \Sigma_{*3} = V_2^T H_1 V_2$, and $\Sigma_{*4} = V_2^T H_2 V_2$, where H_1 and H_2 are diagonal matrix with the jth elements equal to $D(1000, 100, j)$ and $D(1000-\epsilon, 100-\epsilon, j)$ respectively. By constructing precision matrices in this way, each Ω_{*c} is dense and relatively ill-conditioned.

We set all elements of $\mu_{*1} = 15\log(p)/p$, $\mu_{*2} = 7.5\log(p)/p$, $\mu_{*3} = 0$, and $\mu_{*4} = -7.5\log(p)/p$. We investigate the settings $(p, \epsilon) \in \{20, 50\} \times \{1.0\}$. We also tried $\epsilon \in \{0.05, 5.0, 9.0\}$, but results were effectively the same. Similar data generating models were used by Price et al. (2015).

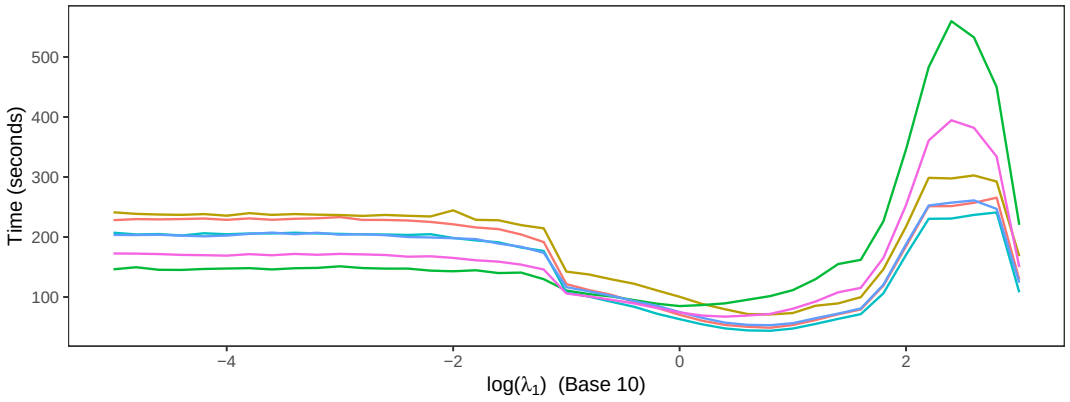
Under this data generating model we expect CRF to perform well in detecting the clustered structures, while a method such as RDA could also perform well since the resulting



(a) $p = 20$



(b) $p = 50$



(c) $p = 100$

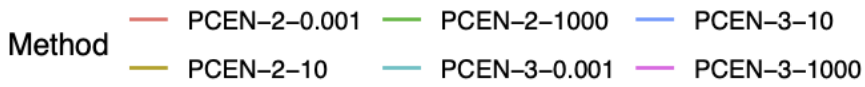
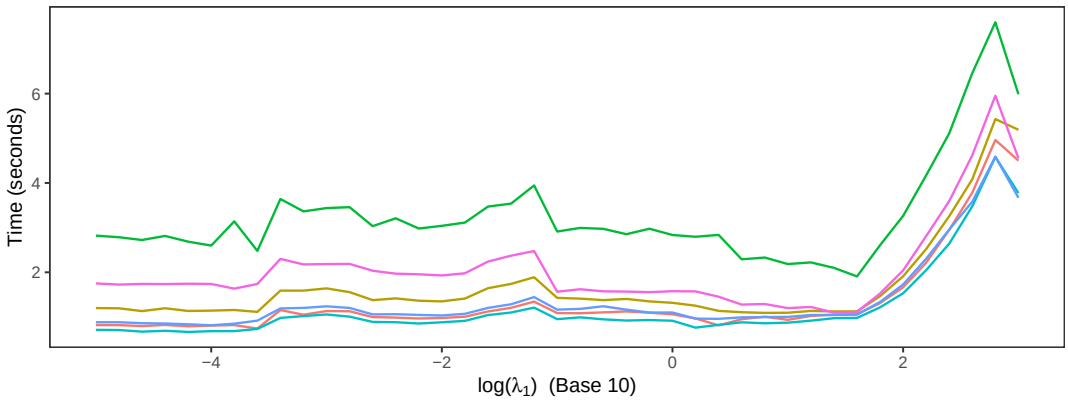
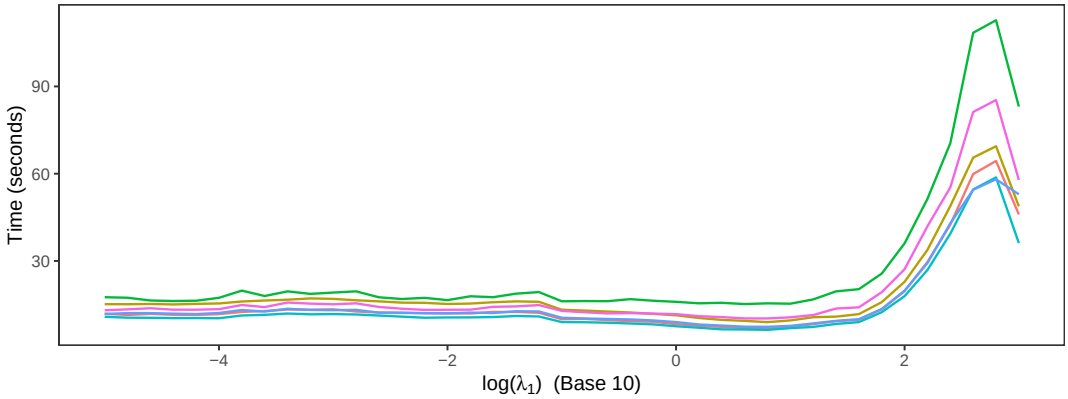


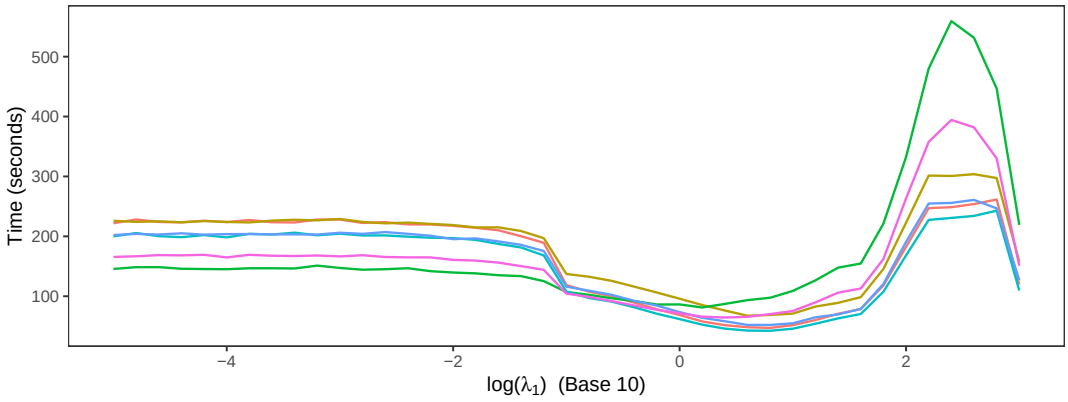
Figure 2: Results for the simulation setting described in Section 4.2 in the manusript (a) considers $p = 20$, (b) considers $p = 50$, and (c) considers when $p = 100$. Each line represents the average seconds of convergence of 50 replications of the denoted method when λ_2 is fixed, and λ_1 varies.



(a) $p = 20$



(b) $p = 50$



(c) $p = 100$

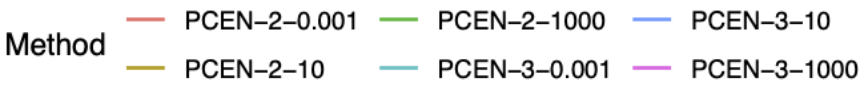
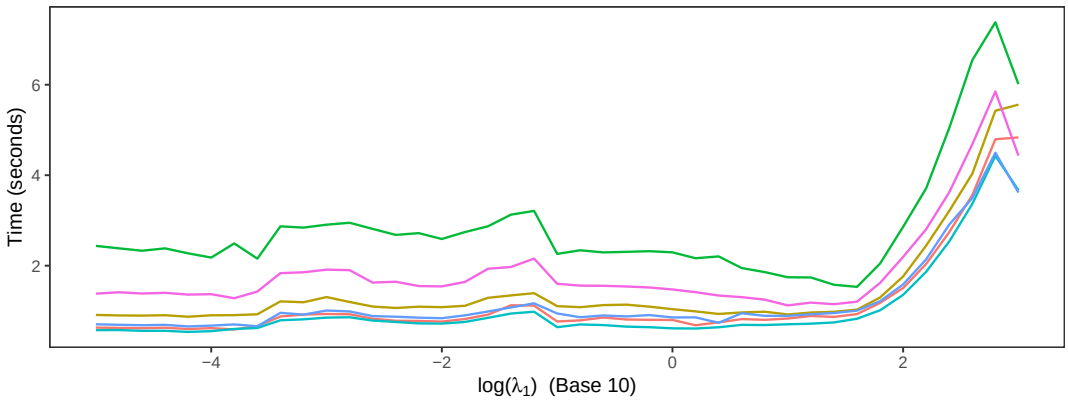
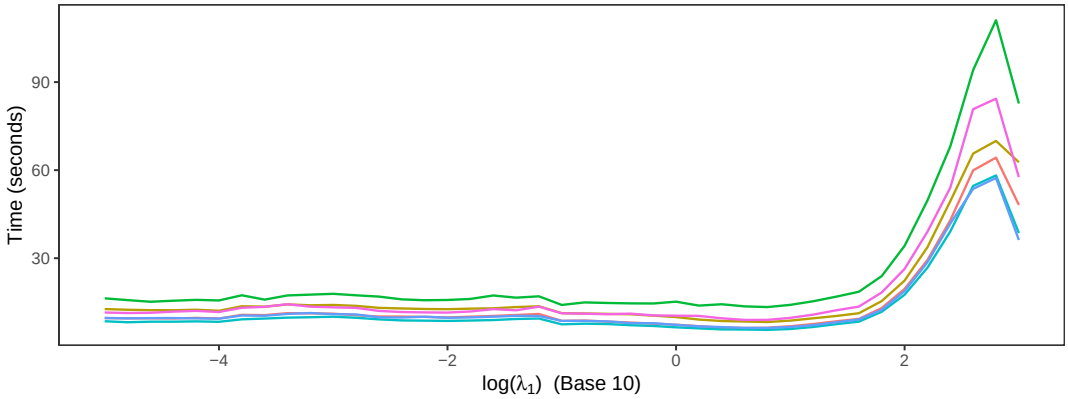


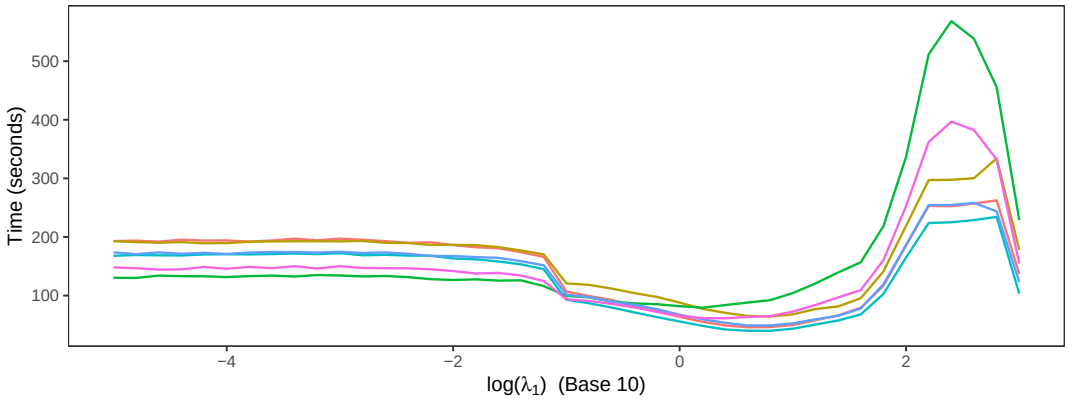
Figure 3: Results for the simulation setting described in Section 4.3 in the manusript (a) considers $p = 20$, (b) considers $p = 50$, and (c) considers when $p = 100$. Each line represents the average seconds of convergence of 50 replications of the denoted method when λ_2 is fixed, and λ_1 varies.



(a) $p = 20$



(b) $p = 50$



(c) $p = 100$

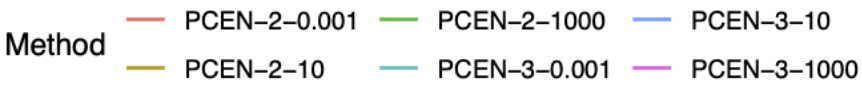
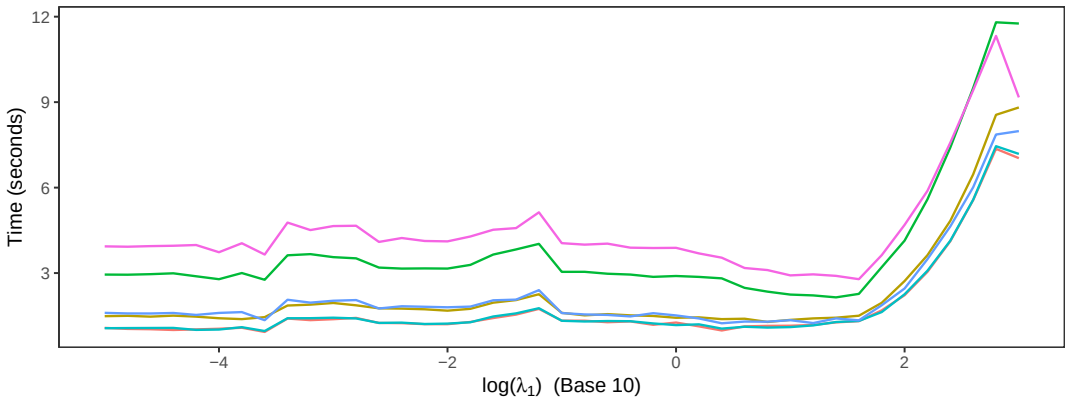
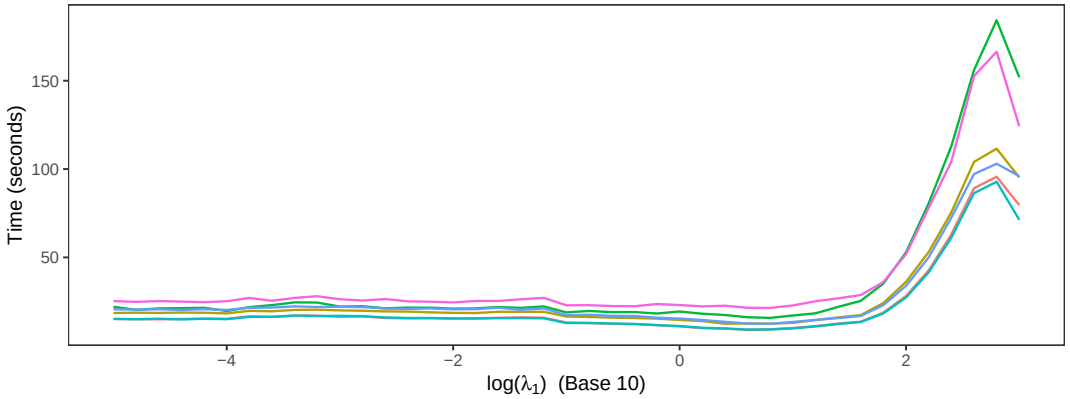


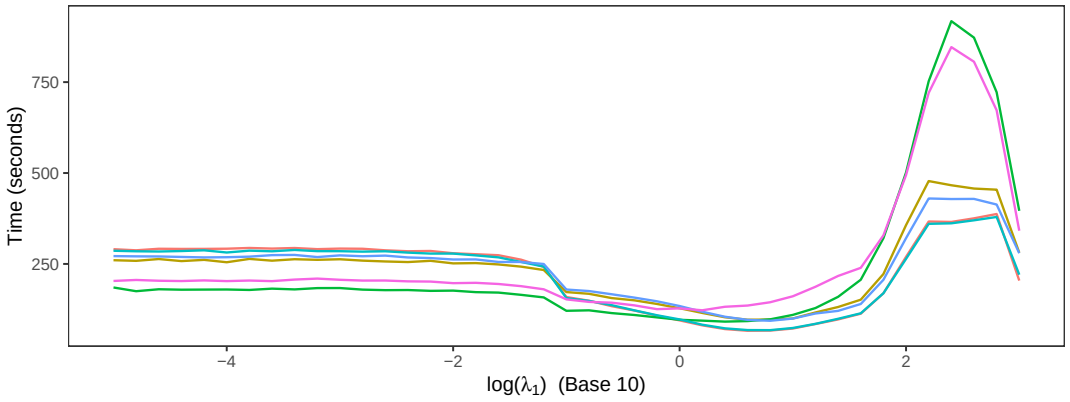
Figure 4: Results for the simulation setting described in Section 3.1 in the supplemental material (a) considers $p = 20$, (b) considers $p = 50$, and (c) considers when $p = 100$. Each line represents the average seconds of convergence of 50 replications of the denoted method when λ_2 is fixed, and λ_1 varies.



(a) $p = 20$



(b) $p = 50$



(c) $p = 100$

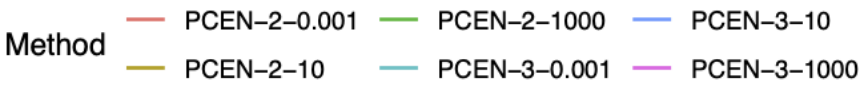
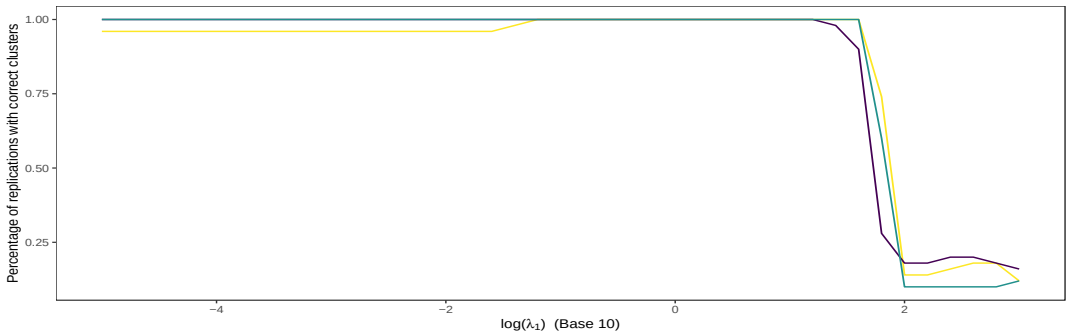
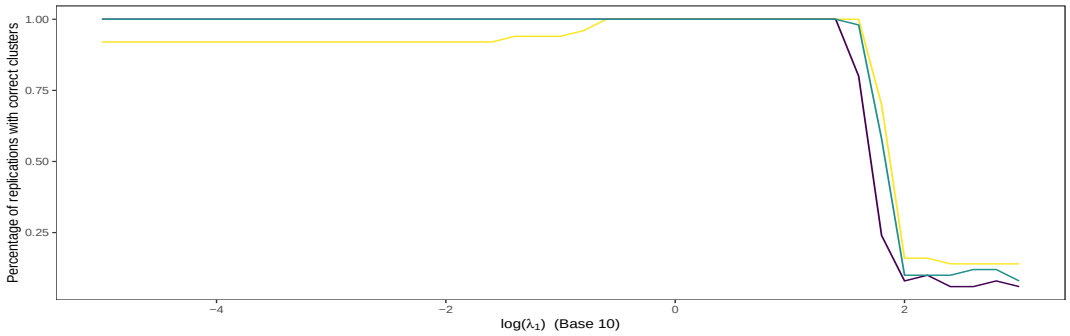


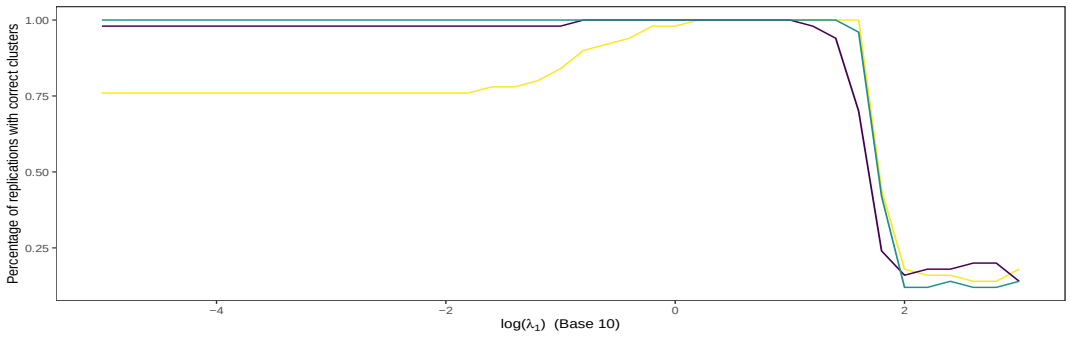
Figure 5: Results for the simulation setting described in Section 4.4 in the manusript (a) considers $p = 20$, (b) considers $p = 50$, and (c) considers when $p = 100$. Each line represents the average seconds of convergence of 50 replications of the denoted method when λ_2 is fixed, and λ_1 varies.



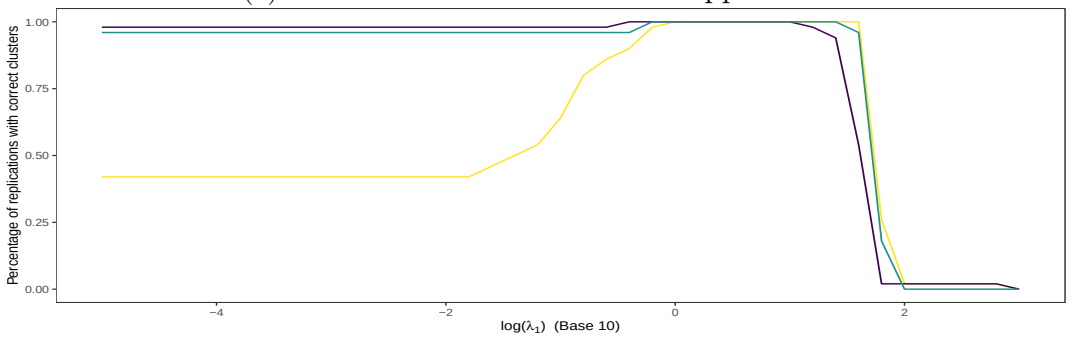
(a) Results from Section 4.2



(b) Results from Section 4.3



(c) Results from Section 3.1 in Supplement



(d) Results from Section 4.4

Figure 6: Results of cluster detection for the graphical model simulations related for PCEN. These results are for $\lambda_2 = 10^3$ for the optimal Q in each simulation.

Table 1: Results of simulation described in Section 3.4 comparing classification error rates and standard errors of CRF, RDA, RF and the two oracle methods for $(p, \epsilon) \in \{20, 50\} \times \{1.0\}$.

	RF	CRF	RDA	Oracle	TC
$p = 20$	0.204	0.022	0.010	0.004	0.006
	(0.008)	(0.003)	(0.002)	(0.000)	(0.003)
$p = 50$	0.185	0.002	0.007	0.000	0.000
	(0.004)	(0.000)	(0.001)	(0.000)	(0.000)

precision matrices are dense.

Table 1 presents a comparison of classification error rates of CRF, RDA, RF and Oracle methods for each p and ϵ. We see the RDA and CRF are competitive for each p and ϵ, while RF has a higher classification error rate in each setting. The results also suggest that as p increases, the variability in classification error rate for CRF, RDA, and the Oracle method decreases. In the $p = 20$ case CRF is able to select the true clusters 12% of the time while for the $p = 50$ case CRF is able to recover the true clusters 100% of the time.

3.5 Effect of sample size on PCEN

In this simulation we investigate the behavior of PCEN-2 using the same data generating model described in Section 4.2 with $n \in \{50, 200\}$, $p \in \{20, 50\}$, and $\lambda_2 \in \{10^{-6}, 10^{-5}, \ldots, 10^{5}, 10^{6}\}$. Each combination is replicated 10 times. The results are presented in Figures 7 and 8, where we display the log-sum of the Forbenious norm squared error and the true positive rate (TPR) compared to the number of non-zero elements selected in all precision matrices for all combinations of n and p. Each line in Figures 7 and represents PCEN-2 with λ_2 fixed and λ_1 varying.

For the considered values of p, we see that as the sample size increases, both estimation and graph recovery improve for all values of λ_2 These results show the effect λ_2 has on the estimation and graph recovery in this setting: namely, large values of the tuning parameter λ_2 led to lowest Frobenius squared norm estimation error, but for models which had slightly higher numbers of nonzero entries.

3.6 Additional Results from Gaussian Graphical Model Simulations

Figures 9, 10, 11 present the results from the simulations described in Sections 4.2, 4.3, and 4.4 respectively, when $p = 20$ and $p = 50$. In each case the results show a similar pattern to the results presented in the manuscript.

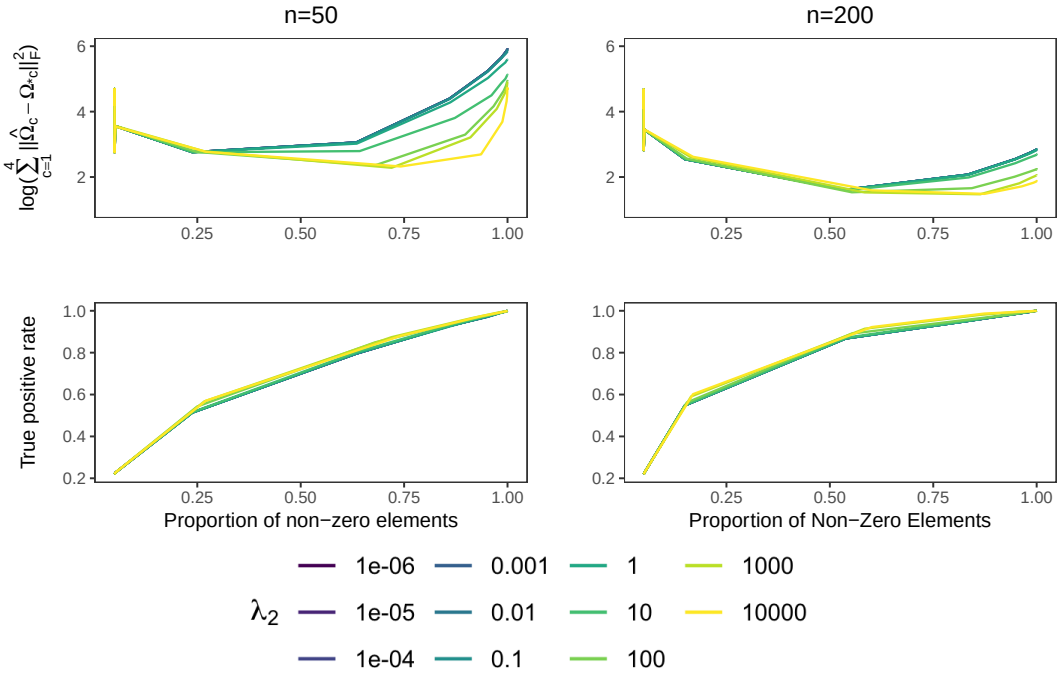


Figure 7: Results for the simulation setting described in Section 3.5 when $p = 20$. Each line represents the average of 10 replications of PCEN when $Q = 2$, λ_2 is fixed, and λ_1 varies.

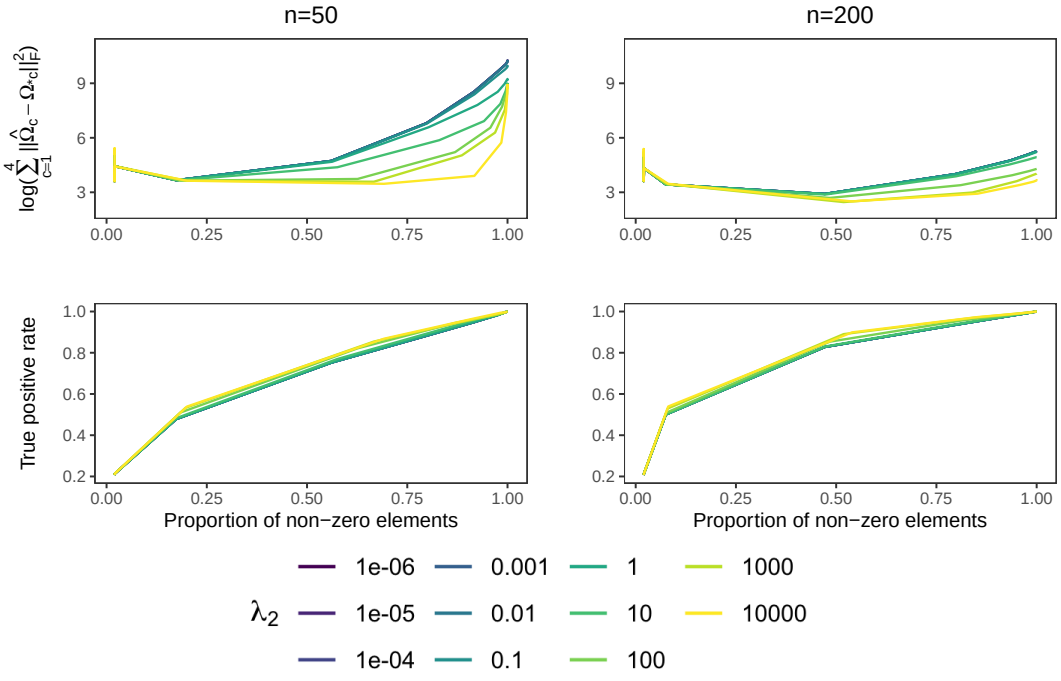


Figure 8: Results for the simulation setting described in Section 3.5 when $p = 50$. Each line represents the average of 10 replications of PCEN when $Q = 2$, λ_2 is fixed, and λ_1 varies.

Table 2: A comparison of network differences produced by Joint Graphical Lasso for the Pulmonary Hypertension Patients Data. The values in the table are the number of edges that are present only in IPAH, SPAH, or are present in both.

	IPAH	SS w/o PH	IPAH and SS w/o PH	All Groups	Total
Graphical Lasso	1453	22	5	1110	2037

3.7 Access to Code and Data

The code used to produce the simulations and data analysis examples can be found in a public github repository at `https://github.com/bprice2652/cluster_fusion_precision`.

4 Gene Expression from Pulmonary Hypertension Patients

Figure 12 presents the corresponding network structure found using PCEN with Q=2 under the setting described in Section 6.1 in the manuscript. Figure 13 and 14 correspond to the network structures found using the cooporative lasso and graphical lasso respectively.

5 Complete Algorithms

The complete algorithms for CRF and PCEN can be found in Algorithm 1 and Algorithm 2 respectively. We note our implementation of PCEN described in Algorithm 2 uses backtracking.

References

Banerjee, O., Ghaoui, L. E., and d'Aspermont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516.

Combettes, P. L. and Wajs, V. R. (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Model Simulation*, 4(4):1168–1200.

Price, B. S., Geyer, C. J., and Rothman, A. J. (2015). Ridge fusion in statistical learning. *Journal of Computational and Graphical Statistics*, 24(2):439–454.

Rolfs, B., Rajaratnam, B., Guillot, D., Wong, I., and Maleki, A. (2012). Iterative thresholding algorithm for sparse inverse covariance estimation. In Pereira, F., Burges, C. J. C.,

Algorithm 1: Blockwise coordinate descent for CRF

Given $Q \in \mathbb{Z}_+$, $\lambda_1 > 0$, $\lambda_2 \geq 0$, $\epsilon > 0$, and initializer $\tilde{\mathbf{\Omega}}^0$, set $t = 1$.

1. $(\tilde{D}_1^t, \ldots, \tilde{D}_Q^t) = \arg\min_{D_1,\ldots D_Q} \left\{ \sum_{q=1}^{Q} \frac{1}{\text{card}(D_q)} \sum_{c,m \in D_q} \|\tilde{\Omega}_c^{t-1} - \tilde{\Omega}_m^{t-1}\|_F^2 \right\}$
2. For all $q \in \{1, \ldots, Q\}$ `# apply ridge fusion algorithm to each cluster`
   a. $\bar{\lambda}_2 = \frac{2\lambda_2}{\text{card}(\tilde{D}_q^t)}$
   b. Repeat for all $m \in D_q$ until $\sum_{m \in D_q} \|\tilde{\Omega}_m^{\dagger} - \tilde{\Omega}_m^{t-1}\|_1 < \epsilon$
      i. $\tilde{\Omega}_m^{\dagger} = \tilde{\Omega}_m^{t-1}$
      ii. Eigendecompose $U\Gamma U' = S_m - \frac{\bar{\lambda}_2}{n_m} \sum_{l \in \tilde{D}_q^t \setminus \{m\}} \tilde{\Omega}_l^{t-1}$
      iii. $\tilde{\lambda}_m = \frac{\lambda_1 + \bar{\lambda}_2(\text{card}(\tilde{D}_q^t) - 1)}{n_m}$
      iv. $\tilde{\Omega}_m^{t-1} = \frac{1}{2\tilde{\lambda}_m} U \{-\Gamma + (\Gamma^2 + 4\tilde{\lambda}_m I_p)^{1/2}\} U'$
   c. For all $m \in D_q$, $\tilde{\Omega}_m^t = \tilde{\Omega}_m^t$
3. If $(\tilde{D}_1^t, \ldots, \tilde{D}_Q^t)$ equals $(\tilde{D}_1^{t-1}, \ldots, \tilde{D}_Q^{t-1})$, terminate. Else, set $t = t + 1$ and return to 1.

Algorithm 2: Blockwise coordinate descent for PCEN

Given $Q \in \mathbb{Z}_+$, $\lambda_1 > 0$, $\lambda_2 \geq 0$, $\epsilon > 0$, and initializer $\tilde{\mathbf{\Omega}}^0$, set $t = 1$.

1. $(\tilde{D}_1^t, \ldots, \tilde{D}_Q^t) = \arg\min_{D_1,\ldots D_Q} \left\{ \sum_{q=1}^{Q} \frac{1}{\text{card}(D_q)} \sum_{c,m \in D_q} \|\tilde{\Omega}_c^{t-1} - \tilde{\Omega}_m^{t-1}\|_F^2 \right\}$
2. For all $q \in \{1, \ldots, Q\}$ `# blockwise coordinate descent for each cluster`
   a. Repeat for all $m \in D_q$ until $\sum_{m \in D_q} \|\tilde{\Omega}_m^{\dagger} - \tilde{\Omega}_m^{t-1}\|_1 < \epsilon$
      i. $\tilde{\Omega}_m^{\dagger} = \tilde{\Omega}_m^{t-1}$
      ii. $\bar{\gamma}_1 = \frac{\lambda_1}{n_m}$
      iii. $\bar{\gamma}_2 = \frac{\lambda_2\{\text{card}(\tilde{D}_q^t) - 1\}}{n_m \text{card}(\tilde{D}_q^t)}$
      iii. $\tilde{S}_m = S_m - \frac{\lambda_2}{n_m \text{card}(\tilde{D}_q^t)} \sum_{c \in \tilde{D}_q^t \setminus \{m\}} \tilde{\Omega}_c^{t-1}$
      iv. Repeat until objective function value from (12) converges `# GEN-ISTA`
         · $\bar{\Omega}_m = \tilde{\Omega}_m^{t-1}$
         · Select step size $\alpha > 0$ by backtracking line search
         · $\tilde{\Omega}_m^{t-1} = \mathcal{S}\left(\bar{\Omega}_m - \alpha \left\{ \tilde{S}_m - \bar{\Omega}_m^{-1} + 2\bar{\gamma}_2 \bar{\Omega}_m \right\}, \alpha\bar{\gamma}_1 \right)$
   b. For all $m \in D_q$, $\tilde{\Omega}_m^t = \tilde{\Omega}_m^t$
3. If $(\tilde{D}_1^t, \ldots, \tilde{D}_Q^t)$ equals $(\tilde{D}_1^{t-1}, \ldots, \tilde{D}_Q^{t-1})$, terminate. Else, set $t = t + 1$ and return to 1.

Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1574–1582. Curran Associates, Inc.

Witten, D. and Tibshirani, R. (2009). Covariance regularized regression and classification for high-dimensional problems. *Journal of Royal Statistical Society, Series B*, 71(3):615–636.

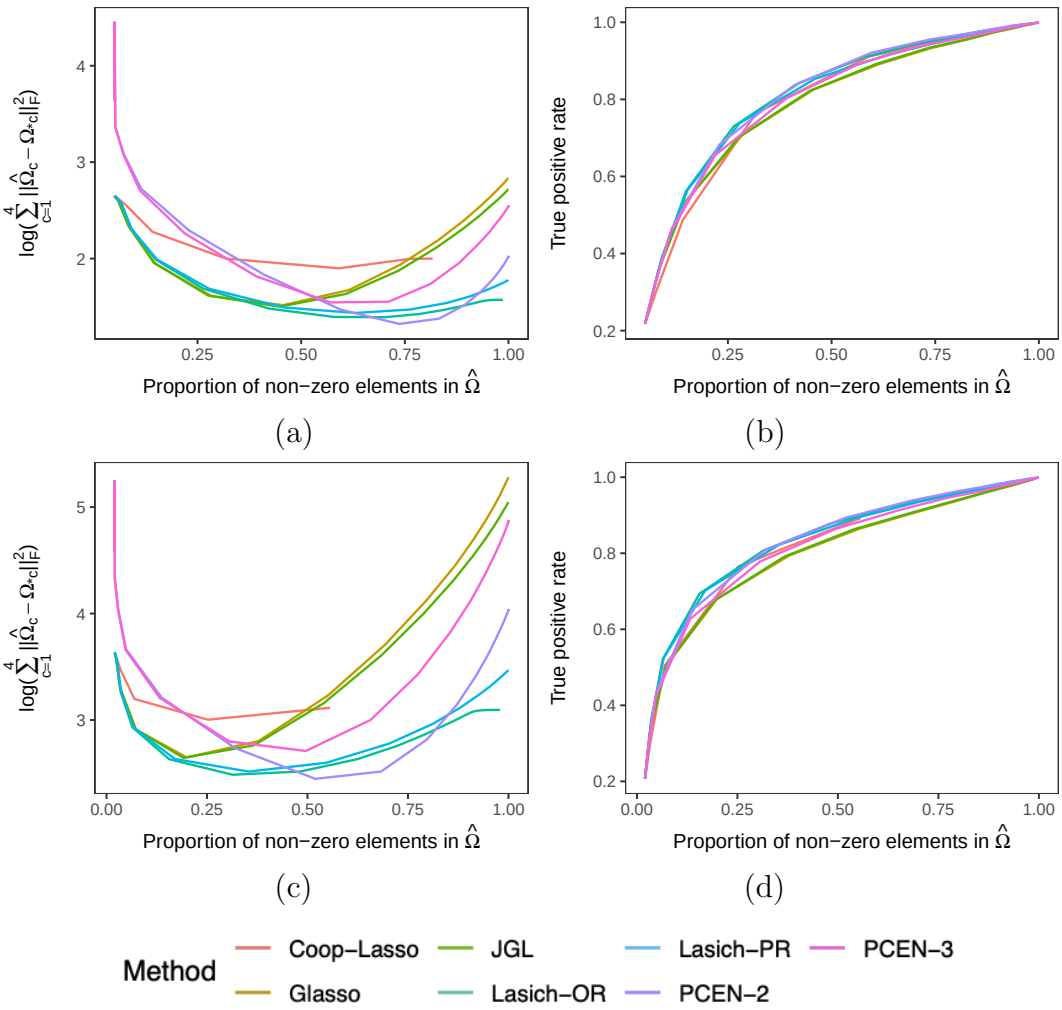


Figure 9: Results for the simulation setting described in Section 4.2. Panels (a,b) present results for the case when $p = 20$, and (c,d) presents the results for when $p = 50$. Each line represents the average of 50 replications of the denoted method when λ_2 is fixed, and λ_1 varies.

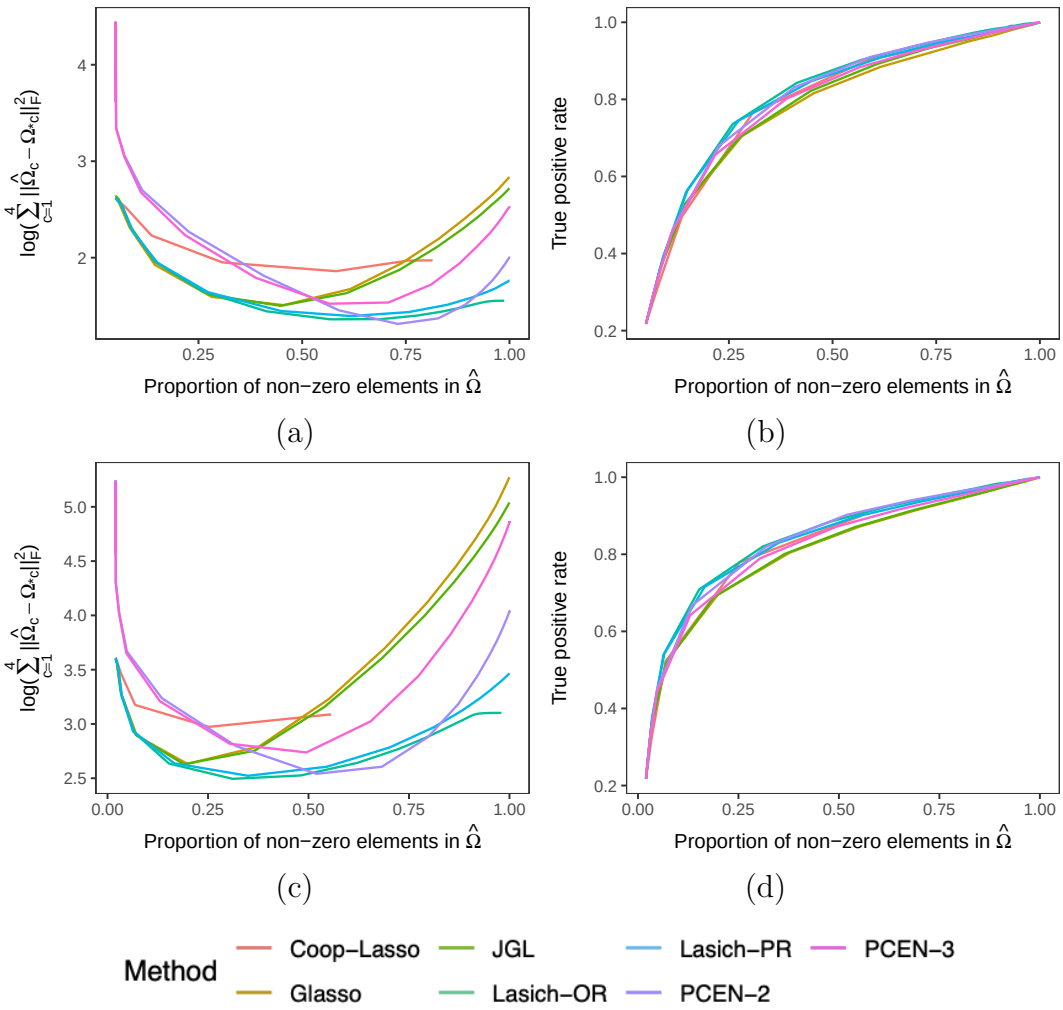


Figure 10: Results for the simulation setting described in Section 4.3. Panels (a,b) present results for the case when $p = 20$, and (c,d) presents the results for when $p = 50$. Each line represents the average of 50 replications of the denoted method when λ_2 is fixed, and λ_1 varies.

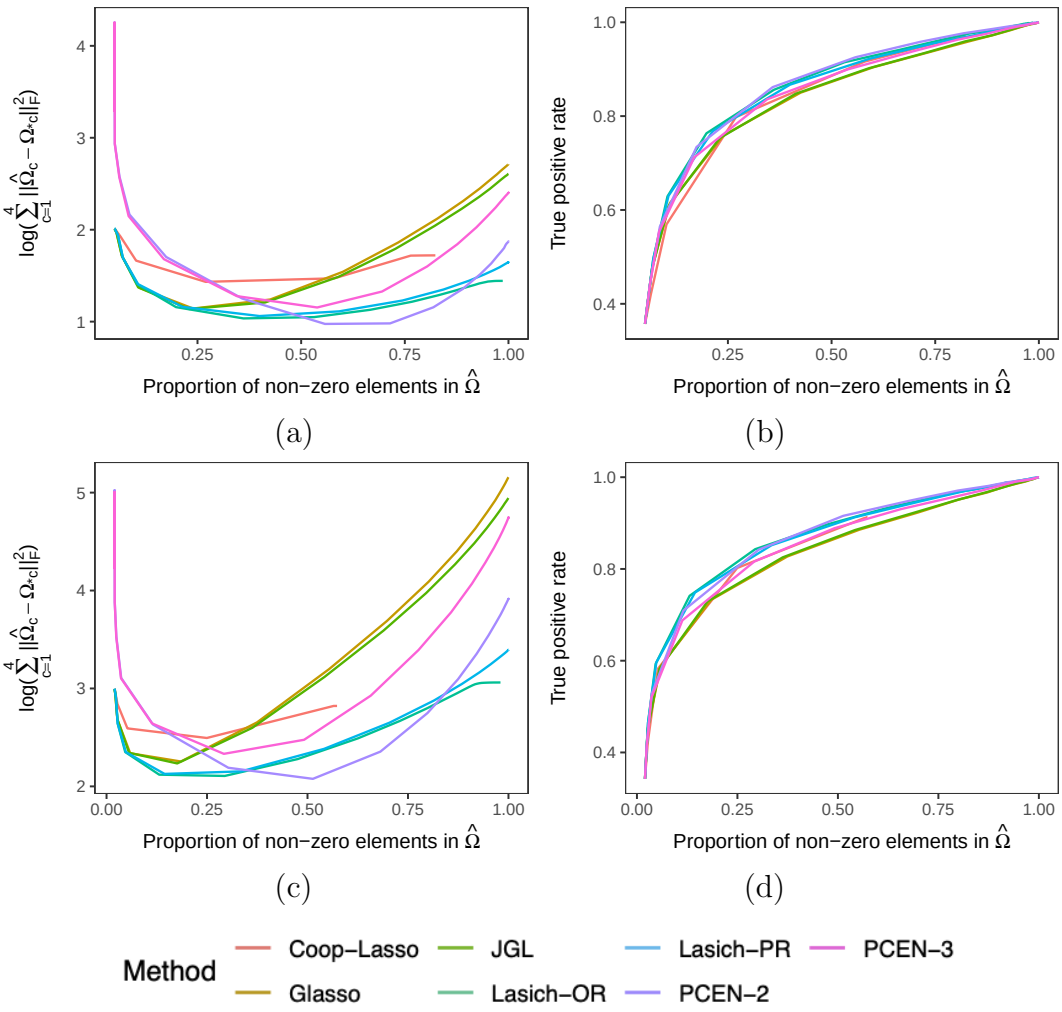


Figure 11: Results for the simulation setting described in Section 4.4. Panels (a,b) present results for the case when $p = 20$, and (c,d) presents the results for when $p = 50$. Each line represents the average of 50 replications of the denoted method when λ_2 is fixed, and λ_1 varies.

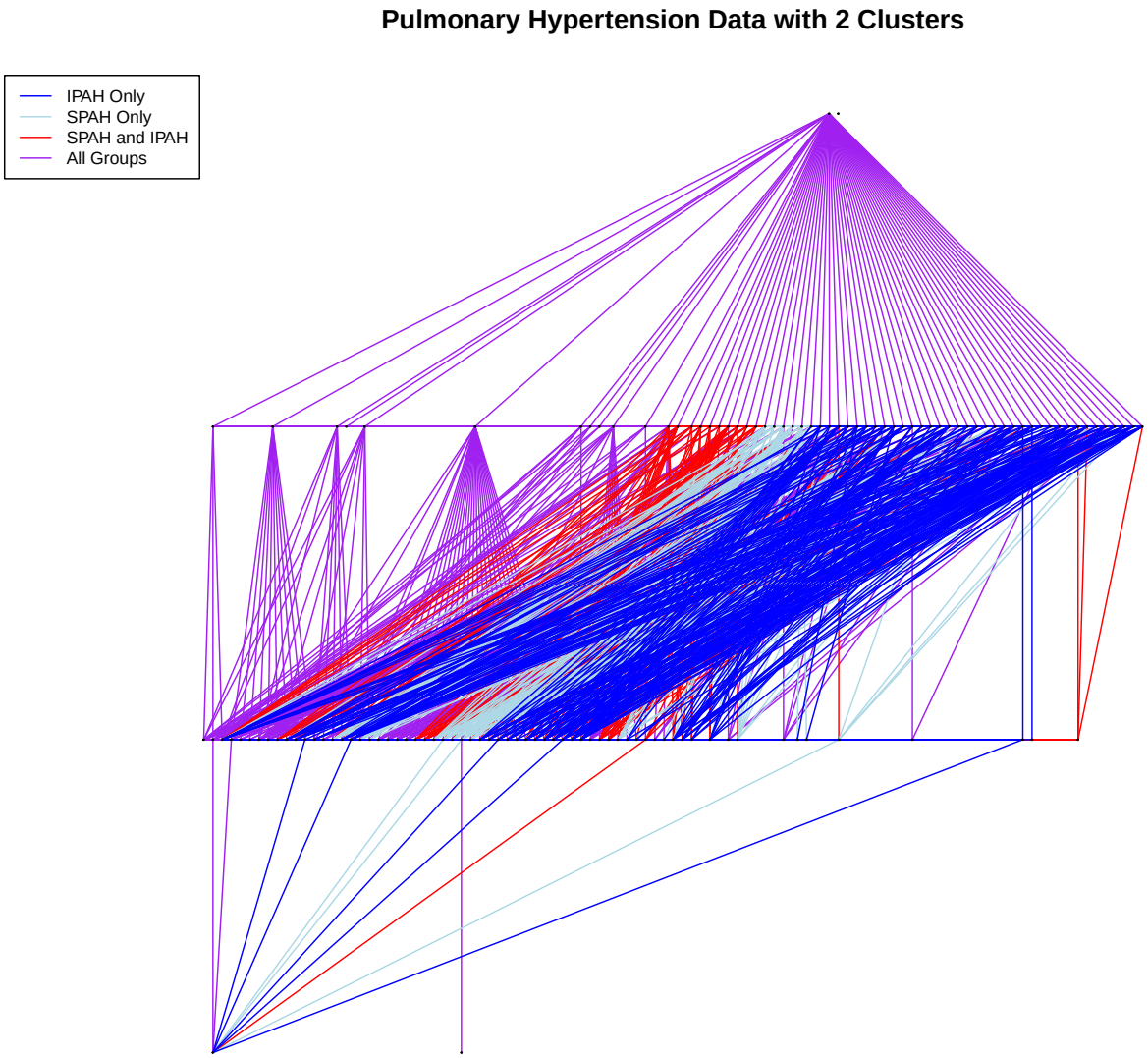


Figure 12: Resulting network comparison from PCEN applied to the Pulmonary Hypertension Patients Data using $Q = 2$ clusters.

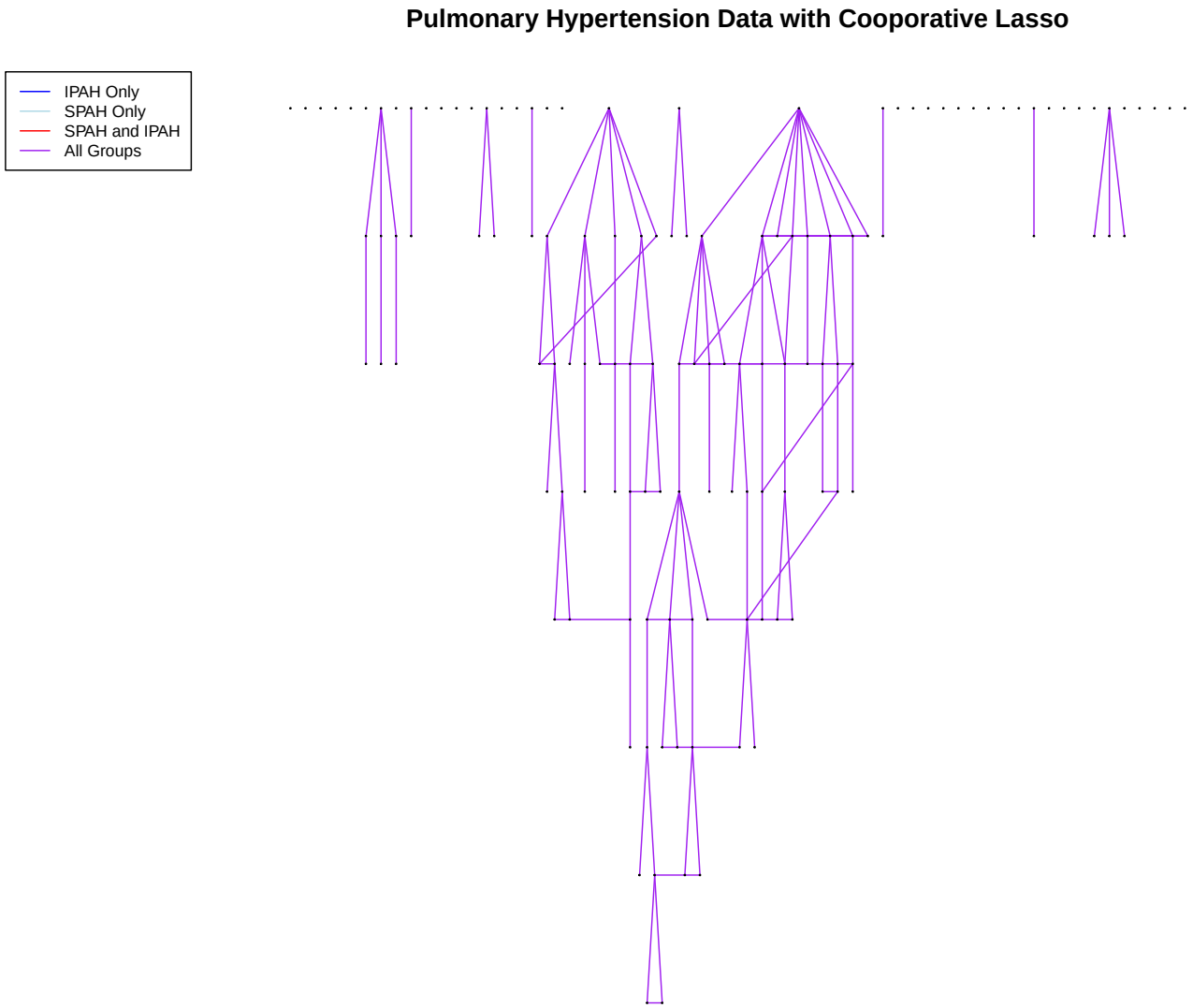


Figure 13: Resulting network comparison Cooporative Lasso applied to the Pulmonary Hypertension Patients Data using largest network fit.

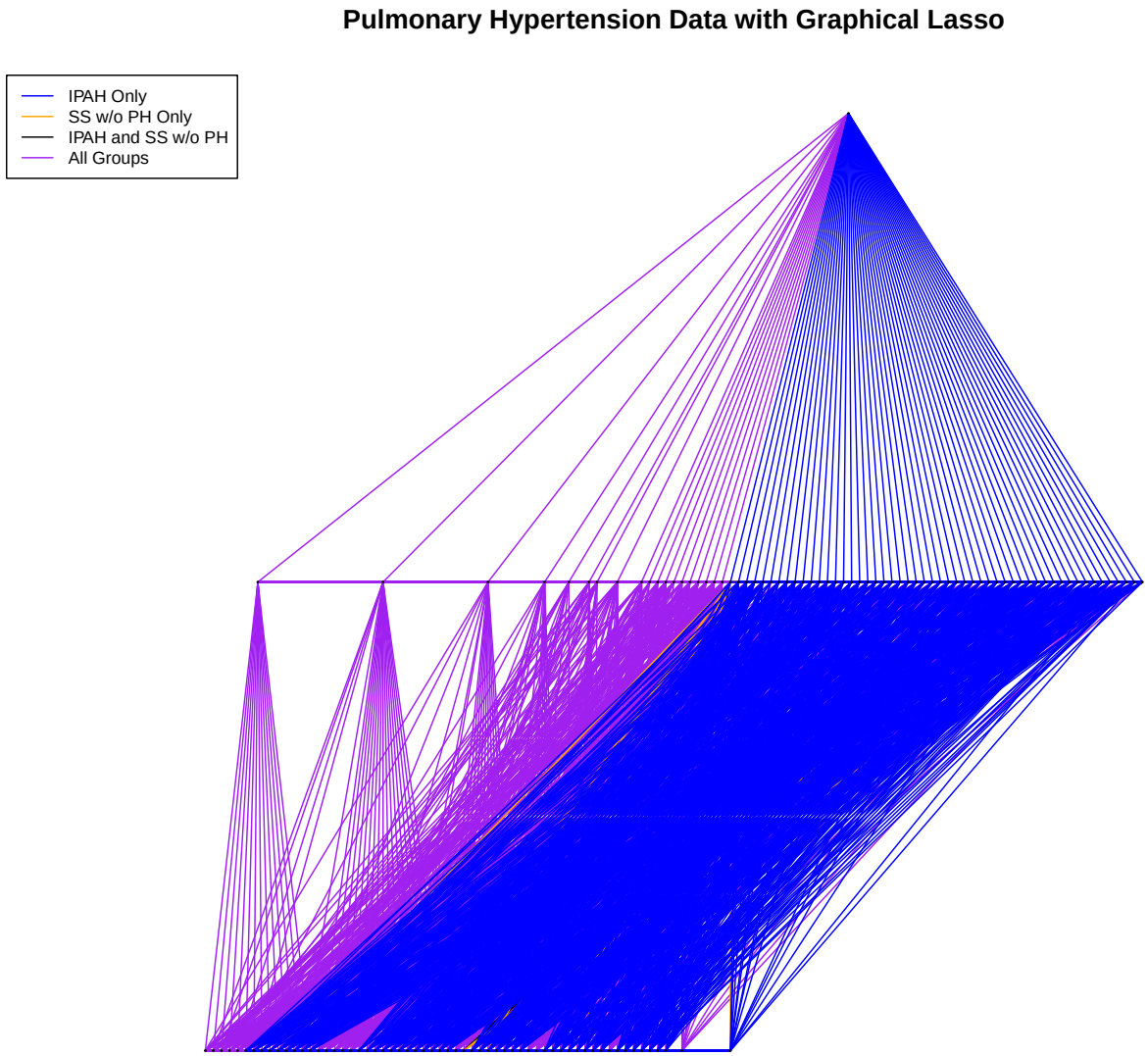


Figure 14: Resulting network comparison from Glasso applied to the Pulmonary Hypertension Patients Data using AIC to select tuning parameters.