Supplementary Materials for "HeteroGGM: an R package for Gaussian graphical model-based heterogeneity analysis" by Mingyang Ren, Sanguo Zhang, Qingzhao Zhang, Shuangge Ma

February 13, 2021

This file contains additional details on the background, methodology, software functions and utilization, and application examples.

1 Background

Heterogeneity analysis has a critical role in the research of many complex diseases, such as cancer (Wolf *et al.*, 2019), COVID-19 (Sun *et al.*, 2021), cardiovascular diseases (Litviňuková *et al.*, 2020), and others. In the literature, there have been multiple definitions of heterogeneity analysis. For example, some studies examine the differences in cells of the same tumor (Turner *et al.*, 2017); and some other studies examine the differences between tumors of different cancer patients (Mathew *et al.*, 2020). In this study, we focus on the scenario under which samples (for example, patients) form subgroups, and a set of variables of interest have different statistical properties across subgroups. Such heterogeneity analysis can be roughly classified as unsupervised (Fang *et al.*, 2017; Hao *et al.*, 2018) and supervised (He *et al.*, 2020; Tang *et al.*, 2020; Wang and Su, 2021). Unsupervised heterogeneity analysis differs from supervised by not including a disease outcome/phenotype. The two types of analysis have different implications and complement but cannot replace each other. With the development of high-throughput profiling techniques, molecular data has been extensively analyzed in heterogeneity analysis. For example, Zhou *et al.* (2009) analyzed the expressions of 300 genes and divided 38 leukemia patients into three subgroups. Guo *et al.* (2010) analyzed 200 genes of small round blue cell tumors of childhood cancer and classified 83 tissue samples into six subgroups. Gao *et al.* (2016) analyzed the expressions of 20 genes and divided 173 core glioblastoma samples into three subgroups. Hao *et al.* (2018) analyzed the expressions of 50 genes and divided 486 glioblastoma samples into four subgroups. More recently, in cancer studies, an alternative source of data that has attracted increasing attention comes from histopathological imaging, which is a byproduct of biopsy – hence enjoying broad availability and high cost-effectiveness. Examples of cancer heterogeneity analysis based on histopathological imaging features include Yu et al. (2017); Choi and Na (2018); Sun et al. (2018); He *et al.* (2020) and others.

Compared to analysis based on simple statistics (mean, variance, correlation, etc.), network-based analysis accounts for interconnections among variables (for example, genes, histopathological imaging features) as well as properties of individual variables, takes a system perspective, and can be more informative. Among the available network analysis techniques, Gaussian graphical model (GGM) has enjoyed high popularity because of its lucid interpretations, computational simplicity, and satisfactory numerical performance. Examples of GGM-based analysis include Wille *et al.*

(2004), which inferred a gene network for isoprenoid biosynthesis in Arabidopsis thaliana and detected modules of closely connected genes and candidate genes for possible cross-talk between the isoprenoid pathways. With gene expression data, Wang *et al.* (2016) conducted the GGM-based analysis of global gene association networks in childhood asthma. In addition, in Wang *et al.* (2016), the networks of synaptic proteins for Alzheimer's patients were constructed and examined, and modules of highly interconnected proteins were detected to elucidate disease physiology. As another example, Fang *et al.* (2017) conducted the GGM-based integrative analysis of multiple cancers and identified significant group-specific interactions of DNA methylation.

1.1 GGM-based heterogeneity analysis

Multiple studies have conducted GGM-based heterogeneity analysis. In the first family of analysis, it is assumed that the heterogeneity structure (that is, which subjects belong to which subgroups) is known, and the focus is on more effectively estimating the GGM parameters. Examples include Guo *et al.* (2011); Danaher *et al.* (2014); Cai *et al.* (2016). Some R packages have been developed to implement these methods, including *JGL* (Danaher *et al.*, 2014), *DiffGraph* (Zhang *et al.*, 2018), and others.

In practical data analysis, heterogeneity structure is usually unknown. To tackle this problem, GGM has been coupled with the mixture modeling technique. In early studies, many focused on the variable selection of mean parameters, either assuming that all subgroups have diagonal precision matrices or considering regularizing individual elements of precision parameters (Zhou *et al.*, 2009). Two representative recent works combine GGM with the truncated penalized fusion technique (with penalty on pairwise differences between subgroups) (Gao *et al.*, 2016) and the group penalization technique (with penalty on elements of precision matrices between subgroups) (Hao *et al.*, 2018). The first approach is realized in the R package *pGMGM*. A common methodological limitation of the aforementioned and several other existing approaches is that the number of subject subgroups is either assumed to be known *a priori* or determined in rather *ad hoc* ways. From an application perspective, the accompanying software programs are often "research oriented" and not friendly or self-contained (meaning that they need to be coupled with other packages for additional analysis, visualization, and summarization).

2 Methods

Very recently, Ren *et al.* (2021) developed a novel GGM-based heterogeneity analysis based on the penalized fusion technique. Penalization has been a popular tool in GGM and other network analyses for regularizing estimation and distinguishing signals (network edges, means, etc.) from noises. Penalized fusion is a relatively recent heterogeneity analysis tool and has multiple notable advantages. Methodologically, the approach developed in Ren *et al.* (2021) is *the first* that can determine the number and structure of subgroups fully data-dependently, thus overcoming the key limitation of the previous studies. To be self-contained and more comprehensive, our package is also designed to realize the approach developed in Zhou *et al.* (2009), which is the most relevant and assumes a known number of subgroups. Below we briefly describe the two methods for the completeness of this article, and refer to the original publications for full details.

2.1 Data settings

With n independent subjects, for subject $i(= 1, \ldots, n)$, p-dimensional measurement $\boldsymbol{x}_i$ – which can be gene expressions, methylation, histopathological imaging features, and others – is available.

These n subjects belong to K_0 subgroups, which have distinct network structures for $\boldsymbol{x}$. For the lth subgroup, assume the distribution:

$$f_l\left(\boldsymbol{x};\boldsymbol{\mu}_l^*,\boldsymbol{\Sigma}_l^*\right)=(2\pi)^{-p/2}\left|\boldsymbol{\Sigma}_l^*\right|^{-1/2}\exp\left\{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}_l^*\right)^\top\left(\boldsymbol{\Sigma}_l^*\right)^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}_l^*\right)\right\},$$

where the mean and covariance matrix are unknown. Overall, $\boldsymbol{x}_i$'s satisfy distribution:

$$f(\boldsymbol{x})=\sum_{l=1}^{K_0}\pi_l^* f_l\left(\boldsymbol{x};\boldsymbol{\mu}_l^*,\boldsymbol{\Sigma}_l^*\right),$$

where the mixture probabilities π_l^*'s are also unknown.

2.2 The penalized fusion-based approach in Ren *et al.* (2021)

Significantly advancing from the previous works, this approach does not assume a known K_0. It defines the penalized objective function:

$$\mathcal{L}(\boldsymbol{\Omega},\boldsymbol{\pi}|\boldsymbol{X}):=\frac{1}{n}\sum_{i=1}^{n}\log\left(\sum_{k=1}^{K}\pi_k f_k\left(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Theta}_k^{-1}\right)\right)-\mathcal{P}(\boldsymbol{\Omega}), \tag{2.1}$$

where $\boldsymbol{X}$ denotes the collection of observed data, $\boldsymbol{\Omega}=(\boldsymbol{\Omega}_1^\top,\cdots,\boldsymbol{\Omega}_K^\top)^\top$, $\boldsymbol{\Omega}_k=\operatorname{vec}(\boldsymbol{\mu}_k,\boldsymbol{\Theta}_k)=(\mu_{k1},\ldots,\mu_{kp},\theta_{k11},\ldots,\theta_{kp1},\ldots,\theta_{k1p},\ldots,\theta_{kpp})\in\mathbb{R}^{p^2+p}$, $\boldsymbol{\Theta}_k=\boldsymbol{\Sigma}_k^{-1}$ is the k-th precision matrix with the ij-th entry θ_{kij}, $\boldsymbol{\pi}=(\pi_1,\cdots,\pi_K)^\top$,

$$\mathcal{P}(\boldsymbol{\Omega})=\sum_{k=1}^{K}\sum_{j=1}^{p}p(|\mu_{kj}|,\lambda_1)+\sum_{k=1}^{K}\sum_{i\neq j}p(|\theta_{kij}|,\lambda_2)+\sum_{k<k'}p\left(\left(\|\boldsymbol{\mu}_k-\boldsymbol{\mu}_{k'}\|_2^2+\|\boldsymbol{\Theta}_k-\boldsymbol{\Theta}_{k'}\|_F^2\right)^{1/2},\lambda_3\right), \tag{2.2}$$

$\|\cdot\|_F$ is the Frobenius norm, and $p(\cdot,\lambda)$ is the base penalty function with tuning parameter $\lambda>0$, which can be Lasso, SCAD, MCP, and others. K is a known constant that satisfies $K>K_0$. In practical data analysis, K can be chosen as a large number to ensure that this is satisfied. Consider the estimate:

$$(\widehat{\boldsymbol{\Omega}},\widehat{\boldsymbol{\pi}})=\underset{\boldsymbol{\Omega},\boldsymbol{\pi}}{\operatorname{argmax}}\mathcal{L}(\boldsymbol{\Omega},\boldsymbol{\pi}|\boldsymbol{X}).$$

Denote $\{\widehat{\boldsymbol{\Upsilon}}_1,\cdots,\widehat{\boldsymbol{\Upsilon}}_{\widehat{K}_0}\}$ as the distinct values of $\widehat{\boldsymbol{\Omega}}$, that is, $\{k:\widehat{\boldsymbol{\Omega}}_k\equiv\widehat{\boldsymbol{\Upsilon}}_l,k=1,\cdots,K\}_{l=1,\cdots,\widehat{K}_0}$ constitutes a partition of $\{1,\cdots,K\}$. Then there are $\widehat{K}_0$ subgroups with estimated mean and precision parameters in $\widehat{\boldsymbol{\Omega}}$. The mixture probabilities can be extracted from $\widehat{\boldsymbol{\pi}}$.

2.3 The penalization approach in Zhou *et al.* (2009)

Under some circumstances, the number of subgroups may be known (based on specific contexts or from other analysis), or only a certain number of subgroups is of interest. To make the software self-contained and more comprehensive, we also develop a function to realize the closely relevant approach developed in Zhou *et al.* (2009) which is designed for known K_0. The estimate is defined as:

$$(\widehat{\boldsymbol{\Omega}}',\widehat{\boldsymbol{\pi}}')=\underset{\boldsymbol{\Omega}',\boldsymbol{\pi}'}{\operatorname{argmax}}\ \frac{1}{n}\sum_{i=1}^{n}\log\left(\sum_{k=1}^{K_0}\pi_k f_k\left(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Theta}_k^{-1}\right)\right)-\sum_{k=1}^{K_0}\sum_{j=1}^{p}p(|\mu_{kj}|,\lambda_1)-\sum_{k=1}^{K_0}\sum_{i\neq j}p(|\theta_{kij}|,\lambda_2),$$

where $\boldsymbol{\Omega}'=(\boldsymbol{\Omega}_1^\top,\cdots,\boldsymbol{\Omega}_{K_0}^\top)^\top$, $\boldsymbol{\pi}'=(\pi_1,\cdots,\pi_{K_0})^\top$, and the other notations are similar to those in Section 2.2.

2.4 Additional penalties

In Ren *et al.* (2021) and Zhou *et al.* (2009), the base penalty function is limited to MCP and Lasso, respectively. To be comprehensive, in the package, we allow all six combinations of the two approaches with Lasso, SCAD, and MCP. This effort significantly enriches analysis with both convex and concave penalties.

2.5 Computational algorithms

In both Ren *et al.* (2021) and Zhou *et al.* (2009), computation is built on the expectation-maximization (EM) technique. In addition, in Ren *et al.* (2021), the alternating direction method of multipliers (ADMM) technique and sparse alternating minimization algorithm (S-AMA) technique are adopted for optimization, which can be somewhat more sophisticated and more effective than that in Zhou *et al.* (2009). In the package, we consistently adopt the same techniques as in Ren *et al.* (2021). Below we provide more details for the approach in Ren *et al.* (2021), and computation for the approach in Zhou *et al.* (2009) is realized in a highly similar manner.

2.5.1 EM algorithm framework

In the t-th step of the EM algorithm, the following function needs to be maximized:

$$E_{\boldsymbol{\gamma}|\boldsymbol{X},\boldsymbol{\Omega}^{(t-1)}}[\mathcal{L}(\boldsymbol{\Omega}|\boldsymbol{X},\boldsymbol{\gamma})] = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\gamma_{ik}^{(t)}\left[\log\pi_k + \log f_k\left(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Theta}_k^{-1}\right)\right] - \mathcal{P}(\boldsymbol{\Omega}), \tag{2.3}$$

where $\mathcal{P}(\boldsymbol{\Omega})$ is defined in (2.2), and $\gamma_{ik}^{(t)}$ can be computed by:

$$\gamma_{ik}^{(t)} = \frac{\pi_k^{(t-1)} f_k\left(\boldsymbol{x}_i;\boldsymbol{\mu}_k^{(t-1)},\left(\boldsymbol{\Theta}_k^{(t-1)}\right)^{-1}\right)}{\sum_{k=1}^{K}\pi_k^{(t-1)} f_k\left(\boldsymbol{x}_i;\boldsymbol{\mu}_k^{(t-1)},\left(\boldsymbol{\Theta}_k^{(t-1)}\right)^{-1}\right)}. \tag{2.4}$$

In the M-step, (2.3) is maximized with respect to $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Theta}_k$. The update of π_k is given by:

$$\pi_k^{(t)} = \frac{1}{n}\sum_{i=1}^{n}\gamma_{ik}^{(t)}. \tag{2.5}$$

For $\boldsymbol{\mu}_k$, maximizing (2.3) with respect to $\{\boldsymbol{\mu}\} = \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ is equivalent to solving:

$$\{\boldsymbol{\mu}^{(t)}\} = \underset{\{\boldsymbol{\mu}\}}{\operatorname{argmin}}\left(\frac{1}{2n}\sum_{i=1}^{n}\sum_{k=1}^{K}\gamma_{ik}^{(t)}\left\{(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top\boldsymbol{\Theta}_k^{(t-1)}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right\} + \mathcal{P}(\boldsymbol{\Omega})\right). \tag{2.6}$$

For this problem, the local quadratic approximation can be adopted to yield an explicit solution at each iteration.

Maximizing (2.3) with respect to $\{\boldsymbol{\Theta}\}$ is equivalent to solving:

$$\{\boldsymbol{\Theta}_k^{(t)}, k = 1, \ldots, K\} = \underset{\{\boldsymbol{\Theta}\}}{\operatorname{argmax}}\left(\sum_{k=1}^{K} n_k\left[\log\{\det(\boldsymbol{\Theta}_k)\} - \operatorname{tr}\left(\widetilde{\boldsymbol{S}}_k\boldsymbol{\Theta}_k\right)\right] - \mathcal{P}(\{\boldsymbol{\Theta}\})\right), \tag{2.7}$$

where $n_k = \sum_{i=1}^{n} \gamma_{ik}^{(t)}$, $\widetilde{\boldsymbol{S}}_k$ is the pseudo sample covariance matrix defined by:

$$\widetilde{\boldsymbol{S}}_k = \frac{\sum_{i=1}^{n} \gamma_{ik}^{(t)} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t)}\right) \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t)}\right)^{\top}}{\sum_{i=1}^{n} \gamma_{ik}^{(t)}},$$

and $\mathcal{P}(\{\boldsymbol{\Theta}\}) = \sum_{k=1}^{K} \sum_{i \neq j} p(|\theta_{kij}|, \lambda_2) + \sum_{k<k'} p\left((\|\boldsymbol{\mu}_k^{(t)} - \boldsymbol{\mu}_{k'}^{(t)}\|_2^2 + \|\boldsymbol{\Theta}_k - \boldsymbol{\Theta}_{k'}\|_F^2)^{1/2}, \lambda_3\right)$.

The solution for (2.7) can be effectively obtained using the ADMM technique. More details are provided in the next subsection. Overall, the EM algorithm is summarized in Algorithm 1.

```
**Algorithm 1** EM algorithm for maximizing (2.1)
**Input**: $\boldsymbol{x}_i, i = 1, \cdots, n$, tuning parameters $\lambda_1, \lambda_2, \lambda_3$, and $K$.
**Output**: Estimated mean vectors and precision matrices.
**Initialization**: Mean vectors $\boldsymbol{\mu}_k^{(0)}$, positive-definite precision matrices $\boldsymbol{\Theta}_k^{(0)}$, and $\pi_k^{(0)}$ obtained using the $K$-means method, for $k = 1, \cdots, K$.
**Repeat for** $t = 1, 2, 3, \ldots$ **as follows**:
    1. E-step: Update the subgroup assignment $\gamma_{ik}^{(t)}$ by (2.4).
    2. M-step: Given $\gamma_{ik}^{(t)}$, update $\pi_k^{(t)}$, $\boldsymbol{\mu}_k^{(t)}$, and $\boldsymbol{\Theta}_k^{(t)}$ by (2.5), (2.6), and (2.7) respectively.
**Until**: $\sum_{k=1}^{K} \left\{ \|\boldsymbol{\mu}_k^{(t)} - \boldsymbol{\mu}_k^{(t-1)}\|_2 \|\boldsymbol{\mu}_k^{(t-1)}\|_2^{-1} + \|\boldsymbol{\Theta}_k^{(t)} - \boldsymbol{\Theta}_k^{(t-1)}\|_F \|\boldsymbol{\Theta}_k^{(t-1)}\|_F^{-1} \right\}$ < a predefined cutoff.
**Return**: Estimate of $\{\boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Theta}_k^{(t)}, \pi_k^{(t)}, k = 1, \ldots, K\}$ at convergence.
```

2.5.2 Update of $\{\boldsymbol{\Theta}\}$ in the EM algorithm

Maximizing (2.3) with respect to $\boldsymbol{\Theta}$ is equivalent to solving:

$$\{\boldsymbol{\Theta}_k^{(t)}, k = 1, \ldots, K\} = \underset{\{\boldsymbol{\Theta}\}}{\operatorname{argmax}} \left(\sum_{k=1}^{K} n_k \left[\log\{\det(\boldsymbol{\Theta}_k)\} - \operatorname{tr}\left(\widetilde{\boldsymbol{S}}_k \boldsymbol{\Theta}_k\right) \right] - \mathcal{P}(\{\boldsymbol{\Theta}\}) \right). \tag{2.8}$$

This can be efficiently achieved using the ADMM technique. More specifically, this optimization can be reformulated as:

$$\underset{\{\boldsymbol{\Theta}, \boldsymbol{\Xi}\}}{\operatorname{argmin}} \left(-\sum_{k=1}^{K} n_k \left[\log\{\det(\boldsymbol{\Theta}_k)\} - \operatorname{tr}\left(\widetilde{\boldsymbol{S}}_k \boldsymbol{\Theta}_k\right) \right] + \mathcal{P}(\{\boldsymbol{\Xi}\}) \right), \tag{2.9}$$

subject to the constraint that $\boldsymbol{\Xi}_k = \boldsymbol{\Theta}_k, k = 1, \cdots, K$ as well as the positive definiteness constraint, where $\{\boldsymbol{\Xi}\} = \boldsymbol{\Xi}_1, \ldots, \boldsymbol{\Xi}_K$, and $\boldsymbol{\Xi}_k = (\xi_{kij})_{1 \leqslant i,j \leqslant p}$. The scaled augmented Lagrangian form for this problem is given by:

$$\begin{aligned} \mathcal{Q}_\kappa(\{\boldsymbol{\Theta}\}, \{\boldsymbol{\Xi}\}, \{\boldsymbol{\Psi}\}) = & -\sum_{k=1}^{K} n_k \left[\log\left\{\det(\boldsymbol{\Theta}_k)\right\} - \operatorname{tr}\left(\widetilde{\boldsymbol{S}}_k \boldsymbol{\Theta}_k\right) \right] + \mathcal{P}(\{\boldsymbol{\Xi}\}) \\ & + \frac{\kappa}{2} \sum_{k=1}^{K} \|\boldsymbol{\Theta}_k - \boldsymbol{\Xi}_k + \boldsymbol{\Psi}_k\|_{\mathrm{F}}^2 - \frac{\kappa}{2} \sum_{k=1}^{K} \|\boldsymbol{\Psi}_k\|_{\mathrm{F}}^2, \end{aligned} \tag{2.10}$$

where $\{\boldsymbol{\Psi}\} = \{\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_K\}$ are dual variables, and κ is the penalty parameter. The ADMM algorithm for solving (2.9) is summarized in Algorithm 2.

Algorithm 2 ADMM algorithm for solving (2.9)

```
**Input**: Pseudo sample covariance matrices $\widetilde{\boldsymbol{S}}_k, k = 1, \cdots, K$, tuning parameters $\lambda_2, \lambda_3$, and penalty parameter $\kappa$.
**Output**: Estimated precision matrices $\{\boldsymbol{\Theta}_k, k = 1, \ldots, K\}$.
**Initialization**: $\boldsymbol{\Theta}_k^{(0)} = \boldsymbol{I}, \boldsymbol{\Xi}_k^{(0)} = \mathbf{0}, \boldsymbol{\Psi}_k^{(0)} = \mathbf{0}$, for $k = 1, \cdots, K$.
**Repeat for** $m = 1, 2, 3, \ldots$:
    1. For $k = 1, \ldots, K$, update $\boldsymbol{\Theta}_k^{(m)}$ by solving
       $\underset{\{\boldsymbol{\Theta}\}}{\operatorname{argmin}} \left( -n_k \left[ \log \left\{ \det \left( \boldsymbol{\Theta}_k \right) \right\} - \operatorname{tr} \left( \widetilde{\boldsymbol{S}}_k \boldsymbol{\Theta}_k \right) \right] + \frac{\kappa}{2} \left\| \boldsymbol{\Theta}_k - \boldsymbol{\Xi}_k^{(m-1)} + \boldsymbol{\Psi}_k^{(m-1)} \right\|_{\mathrm{F}}^2 \right).$
       The solution is given in Witten and Tibshirani (2009).
    2. Update $\{\boldsymbol{\Xi}^{(m)}\}$ by solving:
       $\underset{\{\boldsymbol{\Xi}\}}{\operatorname{argmin}} \left( \frac{\kappa}{2} \sum_{k=1}^{K} \|\boldsymbol{\Xi}_k - \boldsymbol{\mathcal{Z}}_k\|_{\mathrm{F}}^2 + \mathcal{P}(\{\boldsymbol{\Xi}\}) \right) \qquad (2.11)$
       using the S-AMA algorithm, where $\boldsymbol{\mathcal{Z}}_k = \boldsymbol{\Theta}_k^{(m)} + \boldsymbol{\Psi}_k^{(m-1)}$.
    3. Update $\{\boldsymbol{\Psi}^{(m)}\}$ by $\boldsymbol{\Psi}_k^{(m)} = \boldsymbol{\Psi}_k^{(m-1)} + \boldsymbol{\Theta}_k^{(m)} - \boldsymbol{\Xi}_k^{(m)}$, for $k = 1, \ldots, K$.
**Until**: $\sum_{k=1}^{K} \|\boldsymbol{\Theta}_k^{(m)} - \boldsymbol{\Theta}_k^{(m-1)}\|_F \|\boldsymbol{\Theta}_k^{(m-1)}\|_F^{-1} <$ a predefined cutoff.
**Return**: Estimate of $\{\boldsymbol{\Theta}_k^{(m)}, k = 1, \ldots, K\}$ at convergence.
```

2.5.3 Solving (2.11) in the ADMM algorithm

The efficient sparse alternating minimization algorithm (S-AMA) can be used to solve (2.11). The objective function can be rewritten as:

$$\begin{aligned}
&\min_{\{\boldsymbol{\Xi}\}} \frac{\kappa}{2}\sum_{j=1}^{p^2}\left\|\boldsymbol{\xi}_{(j)} - \boldsymbol{z}_{(j)}\right\|_2^2 + \sum_{r\in\mathcal{E}} p\left((\eta_r^{(t)} + \|\boldsymbol{v}_r\|_2^2)^{1/2}, \lambda_3\right) + \sum_{j=1}^{p^2}\sum_{k=1}^{K} p(|\xi_{kj}|, \lambda_2)\cdot I(j\in\mathcal{O}),\\
&\text{s.t. } \mathrm{vec}\boldsymbol{\Xi}_k - \mathrm{vec}\boldsymbol{\Xi}_{k'} - \mathbf{v}_r = 0,
\end{aligned}$$

where $\mathcal{E} = \{(k,k') : 1 \leqslant k, k' \leqslant K\}$, $\eta_r^{(t)} = \|\boldsymbol{\mu}_k^{(t)} - \boldsymbol{\mu}_{k'}^{(t)}\|_2^2$, $\boldsymbol{\xi}_{(j)}, \boldsymbol{z}_{(j)} \in \mathbb{R}^K$ are the j-th columns of $(\mathrm{vec}\boldsymbol{\Xi}_1, \cdots, \mathrm{vec}\boldsymbol{\Xi}_K)^\top$ and $(\mathrm{vec}\boldsymbol{\mathcal{Z}}_1, \cdots, \mathrm{vec}\boldsymbol{\mathcal{Z}}_K)^\top$, respectively, $j = 1, \cdots, p^2$, ξ_{kj} is the k-th element of $\boldsymbol{\xi}_{(j)}$, and $\mathcal{O} = \{j : j \neq d(p+1)+1, d = 0, 1, \cdots, p-1\}$ is the index set of the off-diagonal components of the precision matrices. It is equivalent to minimizing the following augmented Lagrangian function:

$$\begin{aligned}
\mathcal{Q}_{\kappa'}(\{\boldsymbol{\Xi}\}, \boldsymbol{V}, \boldsymbol{\Delta}) =& \frac{\kappa}{2}\sum_{j=1}^{p^2}\left\|\boldsymbol{\xi}_{(j)} - \boldsymbol{z}_{(j)}\right\|_2^2 + \sum_{r\in\mathcal{E}} p\left((\eta_r^{(t)} + \|\mathbf{v}_r\|_2^2)^{1/2}, \lambda_3\right)\\
&+ \sum_{j=1}^{p^2}\sum_{k=1}^{K} p(|\xi_{jk}|, \lambda_2)\cdot I(j\in\mathcal{O}) + \sum_{r\in\mathcal{E}}\langle\boldsymbol{\delta}_r, \mathbf{v}_r - \mathrm{vec}\boldsymbol{\Xi}_k + \mathrm{vec}\boldsymbol{\Xi}_{k'}\rangle\\
&+ \frac{\kappa'}{2}\sum_{r\in\mathcal{E}}\|\mathbf{v}_r - \mathrm{vec}\boldsymbol{\Xi}_k + \mathrm{vec}\boldsymbol{\Xi}_{k'}\|_2^2,
\end{aligned}$$

where κ' is a small penalty parameter. $\boldsymbol{V} = (\mathbf{v}_1, \cdots, \mathbf{v}_{|\mathcal{E}|})$, and $\boldsymbol{\Delta} = (\boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_{|\mathcal{E}|})$. S-AMA minimizes the augmented Lagrangian problem by alternatively solving one block of variables at a time:

$$\begin{aligned}
&\{\boldsymbol{\Xi}^{(s+1)}\} = \underset{\{\boldsymbol{\Xi}\}}{\mathrm{argmin}}\mathcal{Q}_0(\{\boldsymbol{\Xi}\}, \boldsymbol{V}^{(s)}, \boldsymbol{\Delta}^{(s)}),\\
&\boldsymbol{V}^{(s+1)} = \underset{\boldsymbol{V}}{\mathrm{argmin}}\mathcal{Q}_{\kappa'}(\{\boldsymbol{\Xi}^{(s+1)}\}, \boldsymbol{V}, \boldsymbol{\Delta}^{(s)}),\\
&\boldsymbol{\delta}_r^{(s+1)} = \boldsymbol{\delta}_r^{(s)} + \kappa'(\mathbf{v}_r^{(s+1)} - \mathrm{vec}\boldsymbol{\Xi}_k^{(s+1)} + \mathrm{vec}\boldsymbol{\Xi}_{k'}^{(s+1)})\cdot I(\|\mathbf{v}_r^{(s+1)}\|_2 > 0), r \in \mathcal{E}.
\end{aligned}$$

It is noted that AMA differs from ADMM in the update of $\{\boldsymbol{\Xi}\}$. Specifically, AMA solves $\{\boldsymbol{\Xi}\}$ by treating $\kappa' = 0$. The updating implementations for $\{\boldsymbol{\Xi}\}$ and $\boldsymbol{V}$ can both yield closed forms based on mature regularization techniques.

2.5.4 Tuning parameter selection

For selecting the optimal tuning parameter values, we conduct a grid search and optimize an adaptive BIC-type criterion. Detailed information is provided in Ren *et al.* (2021). Simulation in published studies suggests that this approach is computationally affordable and stable.

3 Main functions in the HeteroGGM package

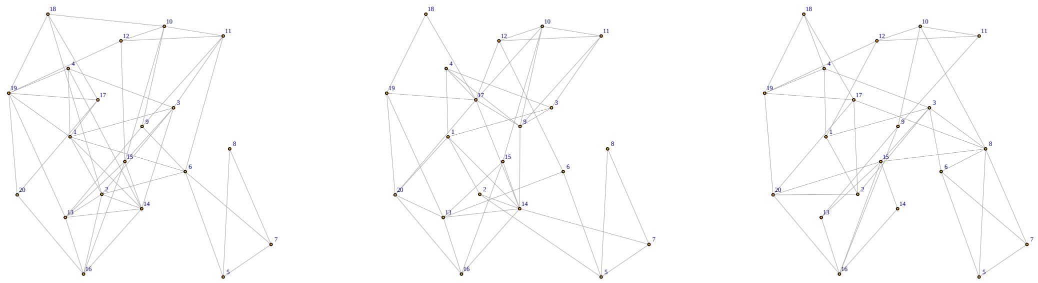
The HeteroGGM package has two main estimation functions, *GMMPF* and *PGGMBC*, corresponding to the approaches described in Sections 2.2 and 2.3, respectively. In the input data matrix, rows correspond to subjects, and columns correspond to variables. With both functions, users can choose from MCP, Lasso, and SCAD as the base penalty. The default is MCP. The output includes the number of subgroups, estimated means and precision matrices, subgrouping memberships (for subjects), and others.

The function *summary-network* can then be called to summarize key characteristics of the resulted network structures, including the numbers of overlapping edges, numbers of edges, information on the connected nodes of a specific node of interest, etc. Based on the output of *summary-network*, the *plot-network* function can be called to visualize the resulted network structures.

These functions are demonstrated below using the example data in the package:

```
> library(HeteroGGM)
> data(example.data)
> K <- 6
> lambda <- genelambda.obo(nlambda1=5,lambda1_max=0.5,lambda1_min=0.1,
+                           nlambda2=15,lambda2_max=1.5,lambda2_min=0.1,
+                           nlambda3=10,lambda3_max=3.5,lambda3_min=0.5)
> res <- GGMPF(lambda, example.data$data, K, penalty = ”MCP”)
> Theta_hat.list <- res$Theta_hat.list
> Mu_hat.list <- res$Mu_hat.list
> opt_num <- res$Opt_num
> opt_Mu_hat <- Mu_hat.list[[opt_num]]
> opt_Theta_hat <- Theta_hat.list[[opt_num]]
> K_hat <- dim(opt_Theta_hat)[3]
> K_hat
[1] 3
> summ <- summary_network(opt_Mu_hat, opt_Theta_hat, example.data$data)
> summ$Theta_summary$overlap
           subgroup 1 subgroup 2 subgroup 3
subgroup 1         98         60         60
subgroup 2         60         88         48
subgroup 3         60         48         84
> va_names <- c(”6”)
> linked_node_names(summ, va_names, num_subgroup=1)
$‘6‘
  linked_node_num linked_node_names
1               1                 1
2               2                 2
3               5                 5
4               7                 7
5               9                 9
6              11                11
> plot_network(summ, num_subgroup = c(1:K_hat), plot.mfrow= c(1,K_hat))
```

The resulted graphical output is as follows.



4 Application examples

4.1 Breast cancer heterogeneity analysis based on gene expression measurements

The heterogeneity of breast cancer has been well acknowledged and examined in multiple molecular studies (Polyak, 2011). Here as a demonstrating example, we analyze the TCGA breast cancer (BRCA) gene expression data. In particular, we focus on the Wnt signaling pathway, which has been established as playing a pivotal role in breast cancer (Louise and Anthony, 2004). The analyzed data is downloaded using the R package *brca.data* (`https://github.com/averissimo/brca.data/releases/download/1.0/brca.data_1.0.tar.gz`). Following published studies, we focus on primary solid tumors and remove genes with missingness, leading to a final dataset of 73 genes and 771 samples. More detailed information on the analyzed data is available from the authors.

We set $K = 10$ (with a much smaller number of subgroups expected), apply the *GGMPF* function, and identify three sample subgroups with sizes 156, 331, and 284, respectively. The default MCP is adopted. The estimated network structures are shown in Figure 1. They have 322, 252, and 68 edges, respectively. Data is also analyzed using the *PGGMBC* function with $K = 3$, for better comparability. The three subgroups have 172, 320, and 279 samples, respectively. The estimated network structures are also shown in Figure 1. They have 402, 302, and 88 edges, respectively. In Table 1, we show the numbers of overlapping edges between subgroups. This information is directly generated by the package. Briefly examining Table 1 suggests that the three subgroups identified using both approaches/functions have significantly different network properties. This is further confirmed by examining interconnections between gene pairs and connectivity of individual genes (details available from the authors). Another observation is that *GGMPF* and *PGGMBC* generate considerably different results, which is also sensible as they have significantly different penalty forms. It is noted that the objective of this study is to deliver convenient software for realizing the existing approaches, whose performance has been extensively examined in the original publications. In penalization studies, a well-adopted strategy is to first determine a rough data structure and then conduct a refit (under the fixed data structure). In our specific context, this means first determining the number of subgroups using GMMPF and then conducting estimation using PGGMBC.

Table 1: Analysis of breast cancer data: numbers of overlapping edges.

	Ren *et al.* (2021)			Zhou *et al.* (2009)		
Subgroup	1	2	3	1	2	3
1	322	102	32	402	128	48
2		252	32		302	40
3			68			88

4.2 LUSC heterogeneity analysis based on histopathological imaging features

Lung squamous cell carcinoma (LUSC) is a major subtype of non-small cell lung cancer (NSCLC) and has demonstrated significant heterogeneity. To demonstrate the broad applicability of the GGM-based heterogeneity analysis techniques and the package, we analyze histopathological imaging features. Briefly, data is downloaded from the TCGA data portal `https://portal.gdc.cancer.gov/projects/TCGA-LUSC`. The automated extraction of imaging features is based on the software CellProfiler, and the detailed pipeline has been described in detail in Zhang *et al.* (2020). The working dataset contains 334 LUSC subjects and 89 imaging features. Brief information on the imaging features is provided in Table 4.

We set $K = 10$, apply the *GMMPF* function (where the default MCP is adopted), and identify six sample subgroups with sizes 43, 47, 52, 63, 77, and 52, respectively. The estimated network structures are shown in Figure 2. They have 248, 222, 388, 158, 218, and 496 edges, respectively. Again, it is observed that different subgroups have significantly different network structures. Data is also analyzed using the *PGGMBC* function with $K = 6$, and the resulted network structures are shown in Figure 3. The six subgroups have 44, 46, 51, 69, 77, and 47 samples, respectively, and their networks have 238, 210, 332, 150, 208, and 470 edges, respectively. As shown in Table 2 and Table 3, observations on the overlapping patterns are similar to those for the breast cancer data.

Table 2: Analysis of LUSC data using the approach in Ren *et al.* (2021): numbers of overlapping edges.

Subgroup	1	2	3	4	5	6
1	248	144	102	104	94	112
2		222	124	124	112	148
3			388	84	128	250
4				158	112	138
5					218	156
6						496

Table 3: Analysis of LUSC data using the approach in Zhou *et al.* (2009): numbers of overlapping edges.

Subgroup	1	2	3	4	5	6
1	238	134	96	100	90	106
2		210	112	122	110	140
3			332	80	114	196
4				150	106	136
5					208	150
6						470

Table 4: LUSC data analysis: Imaging feature numbers and names.

Feature number	Imaging feature names	Feature number	Imaging feature names
1	AreaOccupied-AreaOccupied-Identifyeosinprimarycytoplasm	46	Location-Center-X
2	AreaOccupied-AreaOccupied-identifyhemaprimarynuclei	47	Location-Center-X.1
3	AreaOccupied-Perimeter-Identifyeosinprimarycytoplasm	48	Location-Center-Y
4	AreaOccupied-Perimeter-identifyhemaprimarynuclei	49	Location-Center-Y.1
5	AreaShape-Area	50	Neighbors-AngleBetweenNeighbors-Adjacent
6	AreaShape-Center-X	51	Neighbors-FirstClosestDistance-Adjacent
7	AreaShape-Center-Y	52	Neighbors-FirstClosestObjectNumber-Adjacent
8	AreaShape-MajorAxisLength	53	Neighbors-PercentTouching-Adjacent
9	AreaShape-MaxFeretDiameter	54	Neighbors-SecondClosestDistance-Adjacent
10	AreaShape-Orientation	55	Neighbors-SecondClosestObjectNumber-Adjacent
11	AreaShape-Perimeter	56	ObjectNumber
12	Count-Identifyeosinprimarycytoplasm	57	ObjectNumber.1
13	Count-identifyhemaprimarynuclei	58	Texture-Contrast-ImageAfterMath-3-00
14	Count-Identifyhemasub2	59	Texture-Contrast-ImageAfterMath-3-01
15	Count-identifytissueregion	60	Texture-Contrast-ImageAfterMath-3-02
16	Granularity-1-ImageAfterMath	61	Texture-Contrast-ImageAfterMath-3-03
17	Granularity-1-ImageAfterMath.1	62	Texture-Contrast-maskosingray-3-00
18	Granularity-10-ImageAfterMath	63	Texture-Contrast-maskosingray-3-01
19	Granularity-10-ImageAfterMath.1	64	Texture-Contrast-maskosingray-3-02
20	Granularity-11-ImageAfterMath	65	Texture-Contrast-maskosingray-3-03
21	Granularity-11-ImageAfterMath.1	66	Texture-SumAverage-ImageAfterMath-3-00
22	Granularity-12-ImageAfterMath	67	Texture-SumAverage-ImageAfterMath-3-01
23	Granularity-12-ImageAfterMath.1	68	Texture-SumAverage-ImageAfterMath-3-02
24	Granularity-13-ImageAfterMath	69	Texture-SumAverage-ImageAfterMath-3-03
25	Granularity-13-ImageAfterMath.1	70	Texture-SumAverage-maskosingray-3-00
26	Granularity-14-ImageAfterMath	71	Texture-SumAverage-maskosingray-3-01
27	Granularity-14-ImageAfterMath.1	72	Texture-SumAverage-maskosingray-3-02
28	Granularity-15-ImageAfterMath	73	Texture-SumAverage-maskosingray-3-03
29	Granularity-15-ImageAfterMath.1	74	Texture-SumVariance-ImageAfterMath-3-00
30	Granularity-16-ImageAfterMath	75	Texture-SumVariance-ImageAfterMath-3-01
31	Granularity-16-ImageAfterMath.1	76	Texture-SumVariance-ImageAfterMath-3-02
32	Granularity-2-ImageAfterMath	77	Texture-SumVariance-ImageAfterMath-3-03
33	Granularity-2-ImageAfterMath.1	78	Texture-SumVariance-maskosingray-3-00
34	Granularity-3-ImageAfterMath	79	Texture-SumVariance-maskosingray-3-01
35	Granularity-3-ImageAfterMath.1	80	Texture-SumVariance-maskosingray-3-02
36	Granularity-4-ImageAfterMath	81	Texture-SumVariance-maskosingray-3-03
37	Granularity-4-ImageAfterMath.1	82	Texture-Variance-ImageAfterMath-3-00
38	Granularity-5-ImageAfterMath	83	Texture-Variance-ImageAfterMath-3-01
39	Granularity-6-ImageAfterMath	84	Texture-Variance-ImageAfterMath-3-02
40	Granularity-7-ImageAfterMath	85	Texture-Variance-ImageAfterMath-3-03
41	Granularity-7-ImageAfterMath.1	86	Texture-Variance-maskosingray-3-00
42	Granularity-8-ImageAfterMath	87	Texture-Variance-maskosingray-3-01
43	Granularity-8-ImageAfterMath.1	88	Texture-Variance-maskosingray-3-02
44	Granularity-9-ImageAfterMath	89	Texture-Variance-maskosingray-3-03
45	Granularity-9-ImageAfterMath.1		

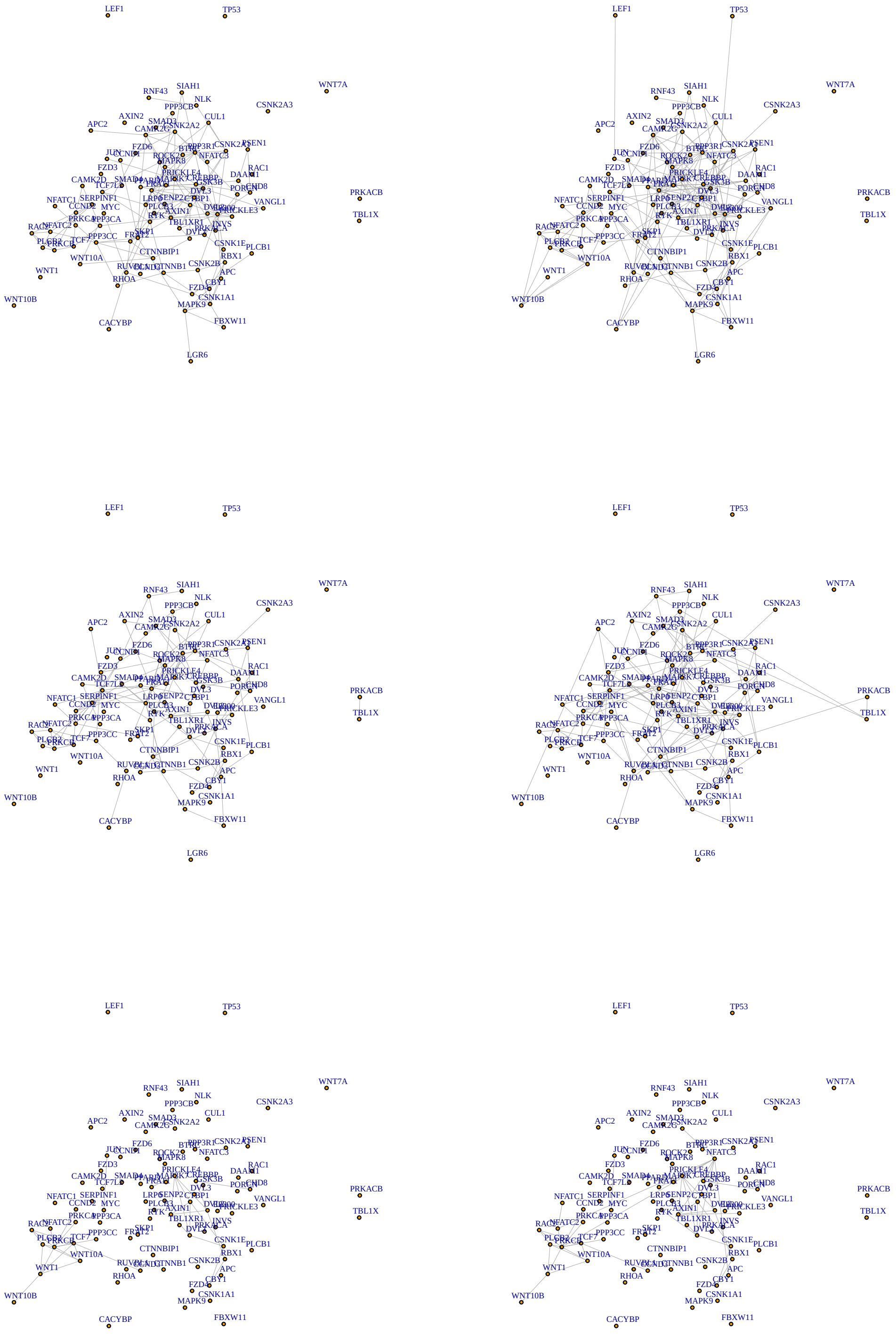


Figure 1: Analysis of breast cancer data using the approaches in Ren *et al.* (2021) (left) and Zhou *et al.* (2009) (right): network structures for individual subgroups.

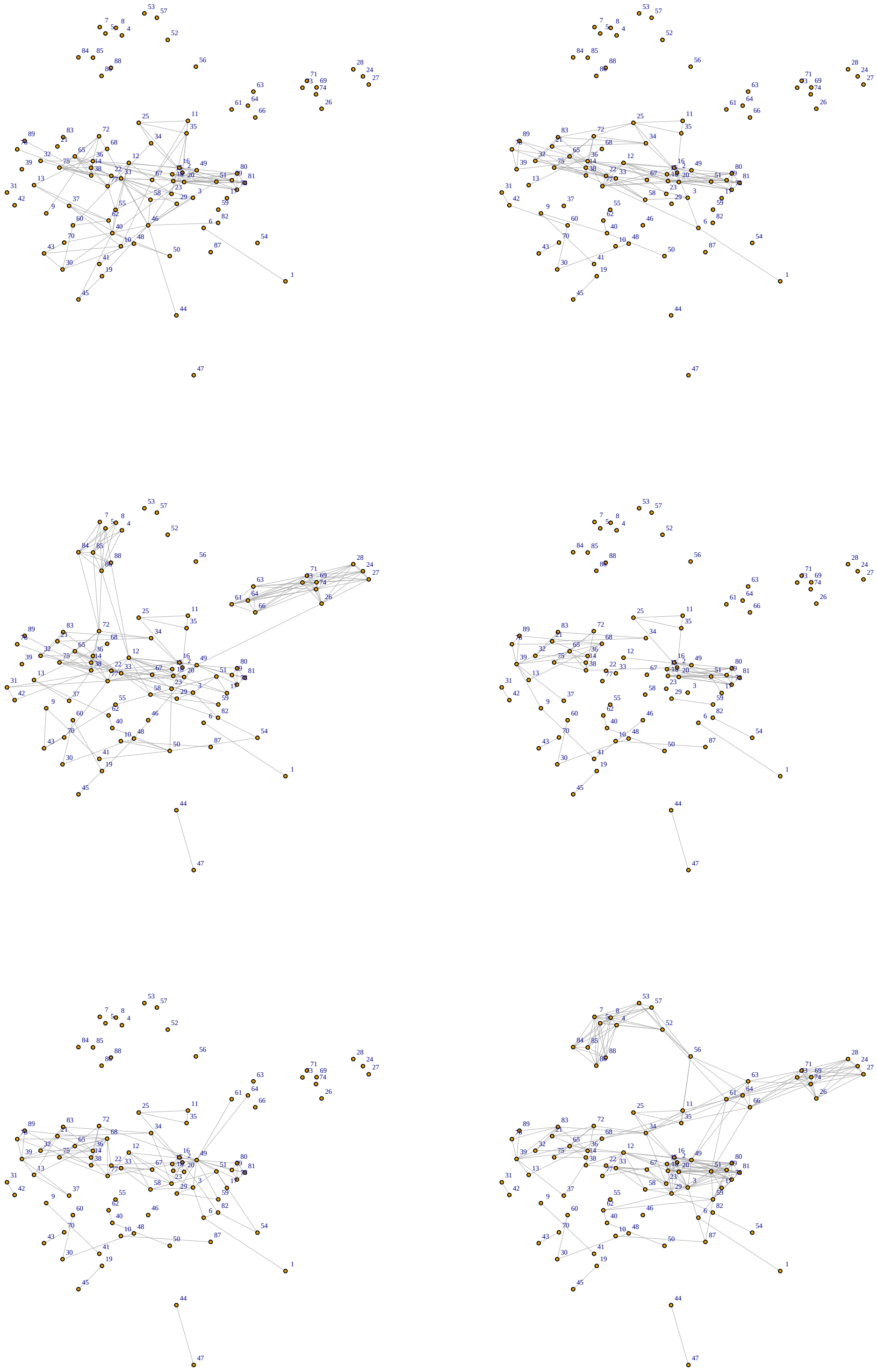


Figure 2: Analysis of LUSC data using the approach in Ren *et al.* (2021): network structures for individual subgroups.

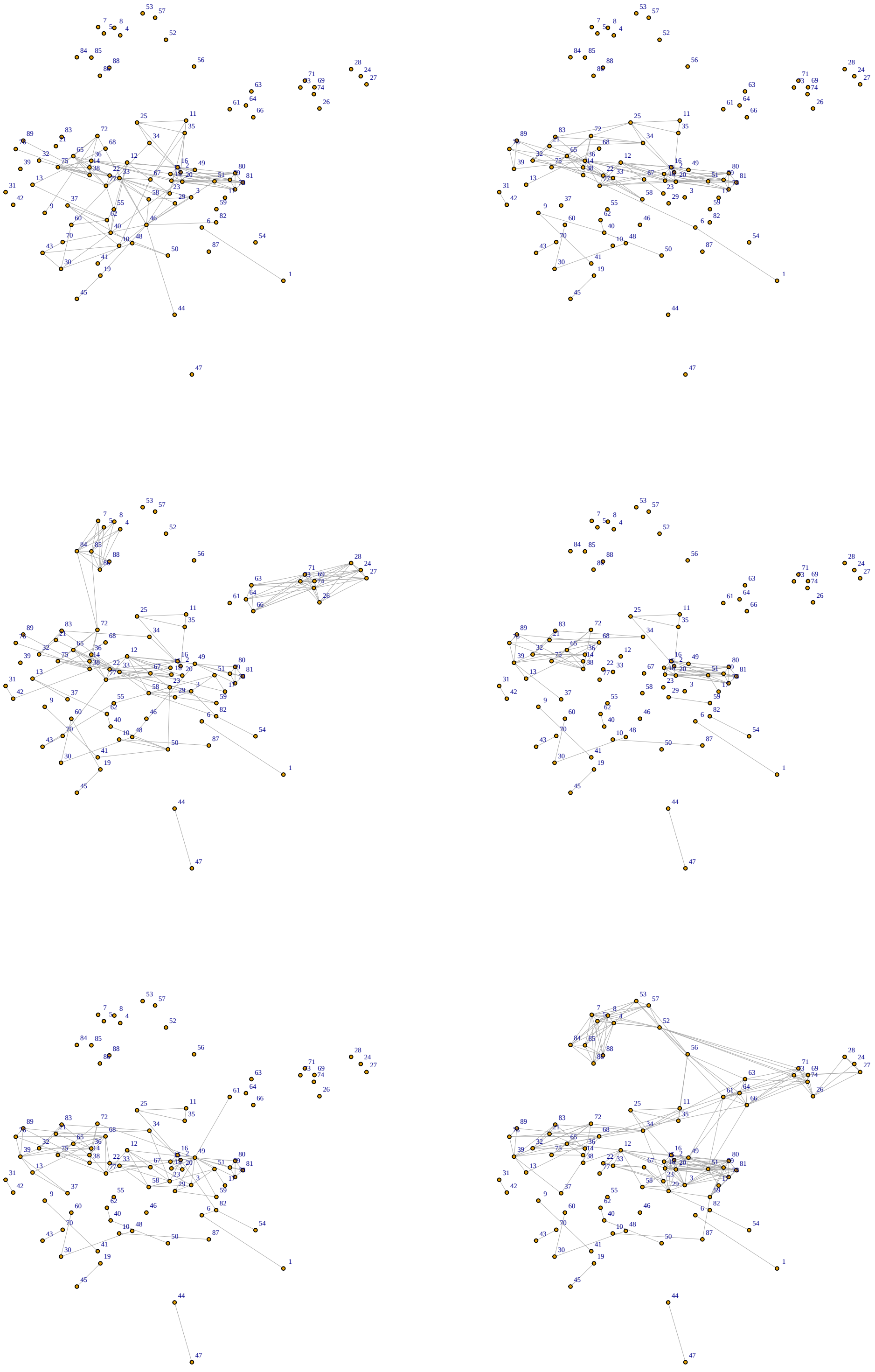


Figure 3: Analysis of LUSC data using the approach in Zhou *et al.* (2009): network structures for individual subgroups.

References

Choi, H. and Na, K. (2018) Integrative analysis of imaging and transcriptomic data of the immune landscape associated with tumor metabolism in lung adenocarcinoma: Clinical and prognostic implications. *Theranostics*, **8**, 1956-1965.

Cai, T., Li, H., Liu, W. and Xie, J. (2016). Joint estimation of multiple high-dimensional precision matrices. *Statistica Sinica*, **26**, 445–464.

Danaher, P., Wang, P. and Witten, D. (2014). The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **76**, 373–397.

Fang, J., Zhang, J., Deng, H. and Wang, Y. (2017). Joint Detection of Associations Between DNA Methylation and Gene Expression From Multiple Cancers. *IEEE journal of biomedical and health informatics*, **22**, 1960-1969.

Gao, C., Zhu, Y., Shen, X. and Pan, W. (2016). Estimation of multiple networks in gaussian mixture models. *Electronic Journal of Statistics*, **10**, 1133-1154.

Guo, J., Levina, E., Michailidis, G. and Zhu, J. (2010). Pairwise variable selection for high-dimensional model-based clustering. *Biometrika*, **66**, 793-804.

Guo, J., Levina, E., Michailidis, G. and Zhu, J. (2011). Joint estimation of multiple graphical models. *Biometrika*, **98**, 1–15.

Hao, B., Sun, W., Liu, Y. and Cheng, G. (2018). Simultaneous clustering and estimation of heterogeneous graphical models. *Journal of Machine Learning Research*, **18**, 7981–8038.

He, B., Zhong, T., Huang, J., et al. (2020). Histopathological imaging-based cancer heterogeneity analysis via penalized fusion with model averaging. *Biometrics*, `https://doi.org/10.1111/biom.13357`.

Litviňuková, M., Talavera-López, C., Maatz, H. et al. (2020). Cells of the adult human heart. *Nature*, **588**, 466–472.

Louise R. and Anthony M. (2004). Wnt Signaling and Breast Cancer. *Cancer Biology and Therapy*, **3**, 36-41.

Mathew, D., Giles, J., Baxter, A. et al. (2020). Deep immune profiling of COVID-19 patients reveals distinct immunotypes with therapeutic implications. *Science*, **369**, 6508.

Polyak K. (2011). Heterogeneity in breast cancer. *The Journal of clinical investigation*, **121**, 3786-3788.

Ren M., Zhang S., Zhang Q. and Ma S. (2021). Gaussian Graphical Model-based Heterogeneity Analysis via Penalized Fusion. *Biometrics*, https://doi.org/10.1111/biom.13426.

Sun, D., Li, A., Tang, B., et al. (2018). Integrating genomic data and pathological images to effectively predict breast cancer clinical outcome. *Computer methods and programs in biomedicine*, **161**, 45-53.

Sun, K., Wang, W., Gao, L. et al. (2021). Transmission heterogeneities, kinetics, and controllability of SARS-CoV-2. *Science*, **371**, 6526.

Tang, X., Xue, F. and Qu, A. (2020). Individualized Multidirectional Variable Selection. *Journal of the American Statistical Association*, `https://doi.org/10.1080/01621459.2019.1705308`.

Turner, K., Deshpande, V., Beyter, D. et al. (2017). Extrachromosomal oncogene amplification drives tumour evolution and genetic heterogeneity. *Nature*, **543**, 122-125.

Wang, T., Ren, Z., Ding, Y. et al. (2016). FastGGM: an efficient algorithm for the inference of gaussian graphical model in biological networks. *PLoS computational biology*, **12**, e1004755.

Wang, W. and Su, L. (2021). Identifying latent group structures in nonlinear panels. *Journal of Econometrics*, **220**, 272-295.

Wille, A., Zimmermann, P., Vranová, E. et al. (2004). Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana. *Genome biology*, **5**, R92.

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

Wolf, Y., Bartok, O., Patkar, S. et al. (2019). UVB-induced tumor heterogeneity diminishes immune response in melanoma. *Cell*, **179**, 219-235.

Yu, K., Berry, G., Rubin, D., et al. (2017). Association of omics features with histopathology patterns in lung adenocarcinoma. *Cell Systems*, **5**, 620-627.

Zhang X., Ou-Yang L., Yang S., et al. (2011). DiffGraph: An R package for identifying gene network rewiring using differential graphical models. *Bioinformatics*, **34**, 1571–1573

Zhang, S., Fan, Y., Zhong, T. and Ma, S. (2020). Histopathological imaging features-versus molecular measurements-based cancer prognosis modeling. *Scientific Reports*, **10**, 15030.

Zhou, H., Pan, W. and Shen, X. (2009). Penalized model-based clustering with unconstrained covariance matrices. *Electronic Journal of Statistics*, **3**, 1473-1496.