Applications of model selection in molecular biology

Mélina Gallopin

June 17, 2021

Biological context

- Gene expression
- Network inference methods

Methods

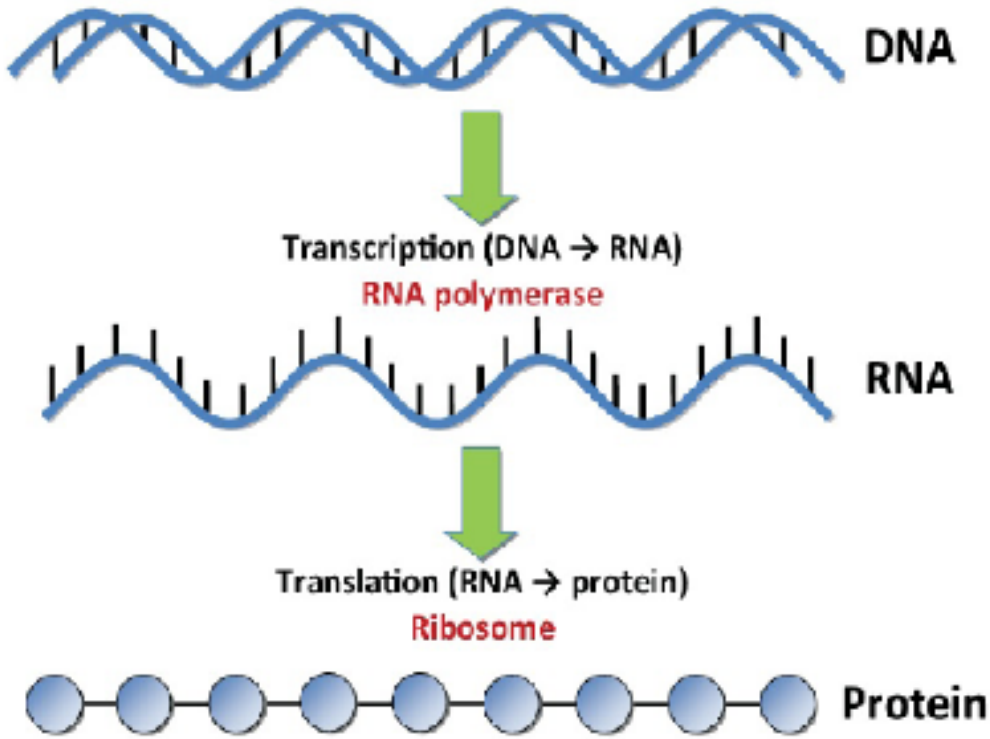
- Undirected graphical models
- Gaussian Graphical Model (GGM)
- GGM in high-dimension : Graphical Lasso

Model selection for detecting modules of genes

- Theoretical aspects
- Practical aspects

Applications on real data

Measuring gene expression



Measuring gene expression using RNA-seq

For a sample :

1. Extraction of RNA
2. Retranscription RNA $\Rightarrow$ DNAc
3. **Lecture of piece of DNAc, called *reads***

GATTACA, GTTTTTAGCTG, TAATTAG

Measuring gene expression using RNA-seq

For a sample :

1. Extraction of RNA
2. Retranscription RNA $\Rightarrow$ DNAc
3. **Lecture of piece of DNAc, called *reads***

$$\textit{GATTACA}, \text{GTTTTTAGCTG}, \text{TAATTAG}$$

4. **Alignment of *reads***

reference genome —TATTTAGCTCTGATTACAATG—

Measuring gene expression using RNA-seq

For a sample :

1. Extraction of RNA
2. Retranscription RNA $\Rightarrow$ DNAc
3. **Lecture of piece of DNAc, called *reads***

$$GATTACA, \text{GTTTTTAGCTG}, \text{TAATTAG}$$

4. **Alignment of *reads***

reference genome —TATTTAGCTCTGATTACAATG—

Measuring gene expression using RNA-seq

For a sample :

1. Extraction of RNA
2. Retranscription RNA $\Rightarrow$ DNAc
3. **Lecture of piece of DNAc, called *reads***

$$GATTACA, \text{GTTTTTAGCTG}, \text{TAATTAG}$$

4. **Alignment of *reads***

reference genome —TATTTAGCTCTGATTACAATG—

Measuring gene expression using RNA-seq

For a sample :

1. Extraction of RNA
2. Retranscription RNA $\Rightarrow$ DNAc
3. **Lecture of piece of DNAc, called *reads***

GATTACA, GTTTTTAGCTG, TAATTAG

4. **Alignment of *reads***

aligned *read* GATTACA
reference genome —TATTTAGCTCTGATTACAATG—

Measuring gene expression using RNA-seq

For a sample :

1. Extraction of RNA
2. Retranscription RNA $\Rightarrow$ DNAc
3. **Lecture of piece of DNAc, called *reads***

GATTACA, GTTTTTAGCTG, TAATTAG

4. **Alignment of *reads***

aligned *read*	TTAGCTC
aligned *read*	GATTACA
reference genome	—TATTTAGCTCTGATTACAATG—

Measuring gene expression using RNA-seq

For a sample :

1. Extraction of RNA
2. Retranscription RNA $\Rightarrow$ DNAc
3. **Lecture of piece of DNAc, called *reads***

GATTACA, GTTTTTAGCTG, TAATTAG

4. **Alignment of *reads***

aligned *read*	GCTCTGAT
aligned *read*	TTAGCTC
aligned *read*	GATTACA
reference genome	—TATTTAGCTCTGATTACAATG—

Measuring gene expression using RNA-seq

For a sample :

1. Extraction of RNA
2. Retranscription RNA $\Rightarrow$ DNAc
3. **Lecture of piece of DNAc, called *reads***

GATTACA, GTTTTTAGCTG, TAATTAG

4. **Alignment of *reads***

aligned *read*	GCTCTGAT
aligned *read*	TTAGCTC
aligned *read*	GATTACA
reference genome	—TATTTAGCTCTGATTACAATG—

5. **Quantification of *Reads***

number of *reads*	45	17685	0	15
reference genome	– gène 1 ———	gène 2 –	gène 3 –	gène 4 —

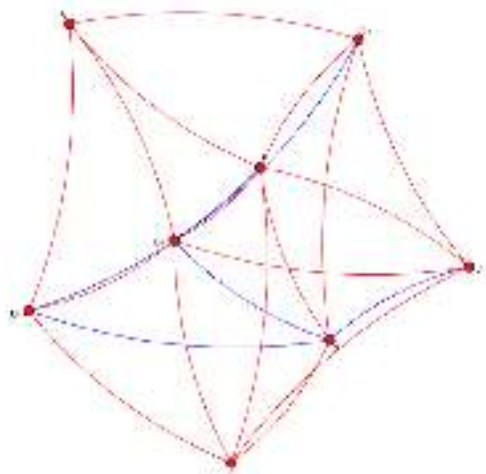
Network inference from expression data

	gene 1	gene 2	gene 3	gene 4	gene 5	...
individual 1	4938	199	2987	0	65	...
individual 2	7530	189	1806	0	29	...
individual 3	2996	201	1752	48	599	...
individual 4	2904	198	2987	0	65	...
individual 5	7670	19931	1837	0	388	...
...	...	...	...	...	...	...

Reconstruct a graph $G = (V, E)$ where

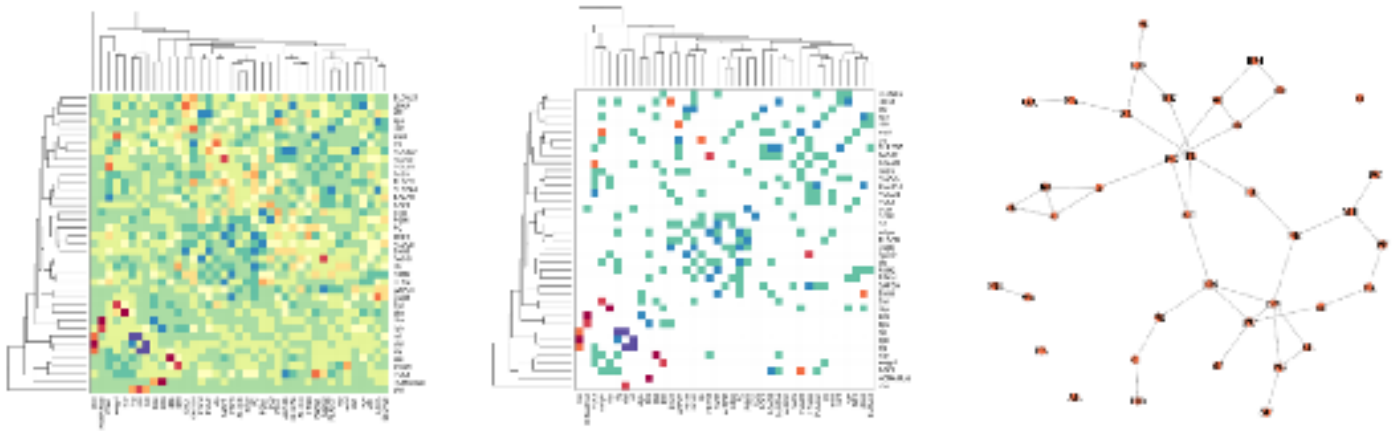
- Vertices $V = \{1, \ldots, p\} \Leftrightarrow$ Random variables (genes)
- Edges $E \Leftrightarrow$ **Direct** dependencies between variables (regulations)

Goal : reconstruct the gene regulatory network



Network inference from gene expression data

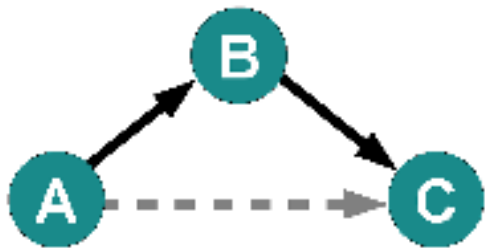
First (naive) approach to build the network: group together similar genes based on pairwise correlations, threshold smallest ones, and build network of correlation (association) network



"Correlations" Thresholding Graph

Figures from Nathalie Villa-Vialaneix

- We want to distinguish between direct and non-direct relationships : if the true underlying is as below, we want to infer an edge between A and B, and B and C, no edge between A and C.



Modeling gene expression data for network inference

	gene 1	gene 2	gene 3	gene 4	gene 5	...
individual 1	4938	199	2987	0	65	...
individual 2	7530	189	1806	0	29	...
individual 3	2996	201	1752	48	599	...
individual 4	2904	198	2987	0	65	...
individual 5	7670	19931	1837	0	388	...
individual 6	2309	18319	8786	20	861	...
individual 7	7398	23101	2237	180	76	...
individual 8	1218	34198	9828	0	65	...
...	...	...	...	...	...	...

y_i: expression level for sample i for $i = 1, \ldots, n$
y^j: expression level for gene j for $j = 1, \ldots, p$
y_{ij}: expression level for gene j and sample i

We observe the expression of p genes $y^1, \ldots, y^p$ and assume that they are realizations of p random variables $Y^1, \ldots, Y^p$.

Network inference methods : an overview (1/2)

(Direct) dependencies networks

- describes marginal dependencies between variables
- two variables Y^j et $Y^{j'}$ independent if we can write their joint distribution as the product of their two marginal distribution (in the following, the letter "p" will represent the corresponding probability density function) :

$$p(y^j; y^{j'}) = p(y^j)p(y^{j'})$$

- related to hierarchical clustering and co-expression networks

Mutual information based networks (*Meyer, 2008*, *Butte, 2000.*)

- The mutual information between two variables is:

$$I(Y^j, Y^{j'}) = \int \int p(y^j, y^{j'}) \log \frac{p(y^j, y^{j'})}{p(y^j)\, p(y^{j'})}\, dy^j dy^{j'}.$$

- takes into account non-linear relationship between variables

Network inference methods : an overview (2/3)

Directed graphical models or Bayesian networks *(Pearl, 1990)*

- Bayesian network are DAG (acyclic directed graph)
- Exemple : consider three random variables Y^1, Y^2, Y^3 and the following factorization of the joint density

$$p(y^1, y^2, y^3) = p(y^1 \mid y^2, y^3) p(y^2 \mid y^3) p(y^3)$$

  The corresponding network has 3 nodes : there is a directed edge from node 3 from to node 2 (due to the factor $p(y^2 \mid y^3)$) and from node 2 and 3 to node 1 (due to the factor $p(y^1 \mid y^2, y^3)$.
- The graph is deduced from the factorisation of the joint density f where $\text{pa}(y^j)$ are the parents of the node j :

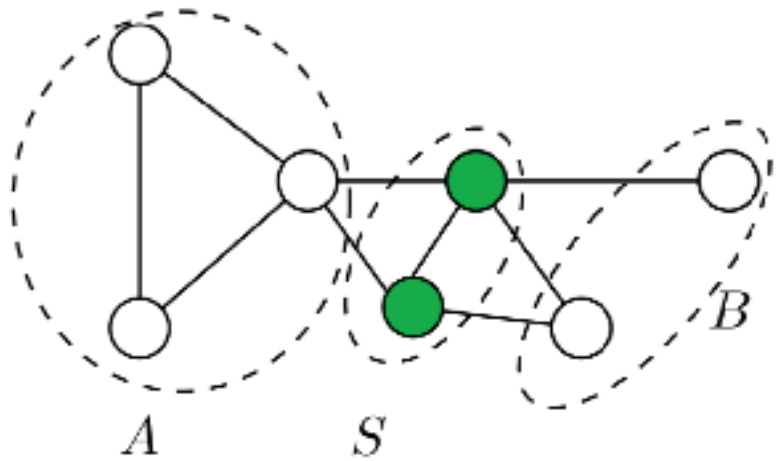
$$p(y^1, \ldots, y^p) = \prod_{j=1}^{p} p(y^k \mid \text{pa}(y^j)).$$

Undirected graphical models *(Whittaker, 1990, Lauritzen, 1996)*

- Conditional dependencies networks, also called Markov networks
- The edges between nodes are non-directed and represent conditional dependencies between variables

A focus on undirected graphical models

We define $Y^S = \{Y^j; j \in S\}$ for any set S of nodes. The vector Y satisfies the Markov property with respect to the graph G if, for any set of nodes S, cutting the graph into two disjoint subsets of nodes A and B, Y^A et Y^B are independent conditionally on Y^S: $p(y^A; y^B \mid y^S) = p(y^A \mid y^S)p(y^B \mid y^S)$.



Hammersley-Clifford Theorem The vector Y satisfies the Markov property with respect to the graph G *iff* the probability distribution density p of the data can be written as follows:

$$p(y^1, \ldots, y^p) = \frac{1}{Z} \prod_{\mathcal{C} \in \mathfrak{C}} \psi_{\mathcal{C}}(y^{\mathcal{C}}).$$

where $\mathcal{C}$ is a fully connected component of the graph, $\mathfrak{C}$ the set of all fully connected component of the graph, ψ_C is a potential function and Z is a partition function (normalization factor).

Special case : the Gaussian Graphical Model (GGM)

- The p variables are assumed to follow Gaussian distributions:
  - $y_i \sim \mathcal{N}_p(0, \Sigma)$ i.i.d. for each individual $i \in \{1, \ldots, n\}$.
  - Σ : covariance matrix of size $p \times p$.
  - $\Theta = \Sigma^{-1}$, the inverse of the covariance matrix, i.e. the **precision matrix**.
  - $\theta_{jj'}$: coefficients of the precision matrix for $(j, j') \in \{1, \ldots, p\}^2$.
- Links between $\theta_{jj'}$ and the partial correlation coefficient $\rho_{jj'}$ between variables j and j' :

$$\rho_{jj'} = \frac{\theta_{jj'}}{\sqrt{\theta_{jj}\theta_{j'j'}}}$$

Special case : the Gaussian Graphical Model (GGM)

- The p variables are assumed to follow Gaussian distributions:
  - $y_i \sim \mathcal{N}_p(0, \Sigma)$ i.i.d. for each individual $i \in \{1, \dots, n\}$.
  - Σ : covariance matrix of size $p \times p$.
  - $\Theta = \Sigma^{-1}$, the inverse of the covariance matrix, i.e. the **precision matrix**.
  - $\theta_{jj'}$: coefficients of the precision matrix for $(j, j') \in \{1, \dots, p\}^2$.
- Links between $\theta_{jj'}$ and the partial correlation coefficient $\rho_{jj'}$ between variables j and j' :

$$\rho_{jj'} = \frac{\theta_{jj'}}{\sqrt{\theta_{jj}\theta_{j'j'}}}$$

To infer the graph, we need to estimate the matrix Θ :

$$y_i \sim \mathcal{N}_3(0, \Sigma)$$

$$\hat{\Sigma} = \begin{pmatrix} 1 & 0.98 & 0.98 \\ 0.98 & 1 & 0.99 \\ 0.98 & 0.99 & 1 \end{pmatrix} \qquad \hat{\Theta} = \begin{pmatrix} 31.3 & -31.6 & 0.86 \\ -31.6 & 145 & -113 \\ 0.86 & -113 & 112 \end{pmatrix}$$

Correlation matrix — Precision matrix

Gaussian Graphical Model in high-dimension

$$y_i \overset{\text{iid}}{\sim} \mathcal{N}_p(0, \Sigma) \text{ pour } i = 1, \ldots, n$$

Each edge in the network $\Leftrightarrow$ non nuls coefficients of $\Theta = \Sigma^{-1}$

In high-dimensional context ("$p > n$")
Maximization on Θ by **Graphical lasso** (Friedman *et al.*, 2008)

$$\log\det(\Theta) - \text{tr}(S\Theta) - \lambda||\Theta||_1$$

with S sample covariance matrix

Gaussian Graphical Model in high-dimension

$$y_i \overset{\text{iid}}{\sim} \mathcal{N}_p(0, \Sigma) \text{ pour } i = 1, \ldots, n$$

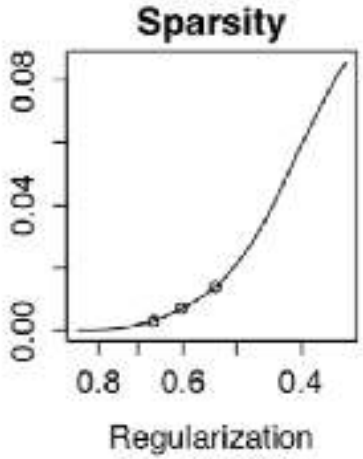
Each edge in the network $\Leftrightarrow$ non nuls coefficients of $\Theta = \Sigma^{-1}$

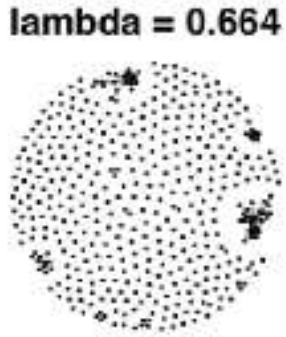
In high-dimensional context ("$p > n$")
Maximization on Θ by **Graphical lasso** (Friedman *et al.*, 2008)

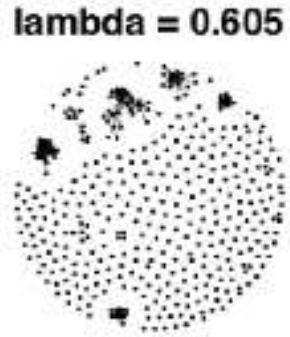
$$\log\det(\Theta) - \text{tr}(S\Theta) - \lambda||\Theta||_1$$

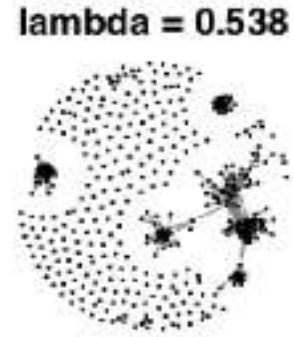
with S sample covariance matrix

Choice of the level of regularization (value of the λ) Bayesian Information Criterion (BIC) or extended BIC (Cheng *et al.*, 2008)









R package **huge** (Liu *et al.*, 2014)

Gaussian Graphical Model in ultra-high dimensional context

Degree of a network d : maximum number of edges adjacent to a node

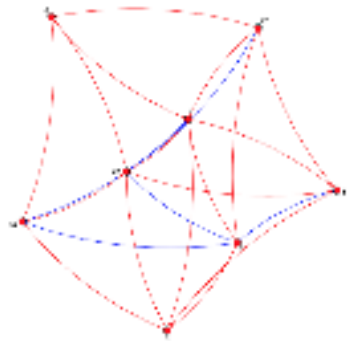
***Ultra-high dimensional* contexts** (Verzelen, 2012)

$$\frac{d \log(\frac{p}{d})}{n} \geq \frac{1}{2}$$

Example: $n = 50, p = 200, d \geq 8 \rightarrow$ network inference is difficult

Gaussian Graphical Model in ultra-high dimensional context

Degree of a network d : maximum number of edges adjacent to a node



***Ultra-high dimensional* contexts** (Verzelen, 2012)

$$\frac{d \log(\frac{p}{d})}{n} \geq \frac{1}{2}$$

Example: $n = 50, p = 200, d \geq 8 \rightarrow$ network inference is difficult

Solutions to reduce the dimension

1. Restrict the number of genes based on external information
2. Select key genes automatically

A property of the graphical lasso algorithm.

Block Diagonal Screening Rule for the glasso (Mazumder et Hastie, 2012)

For a fixed regularization parameter λ, S sample covariance matrix

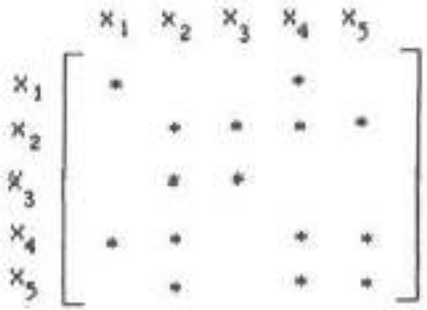
Step 1 Thresholding of $|S|$ to the level $\lambda \Rightarrow$ block structure

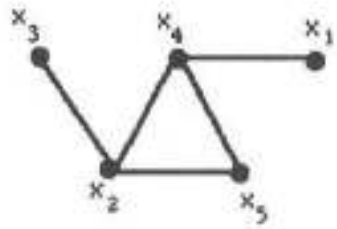
Step 2 Graphical lasso with regularization parameter λ in each block

This rule gave rise to new algorithms Cluster graphical lasso (Tan et al., 2015)

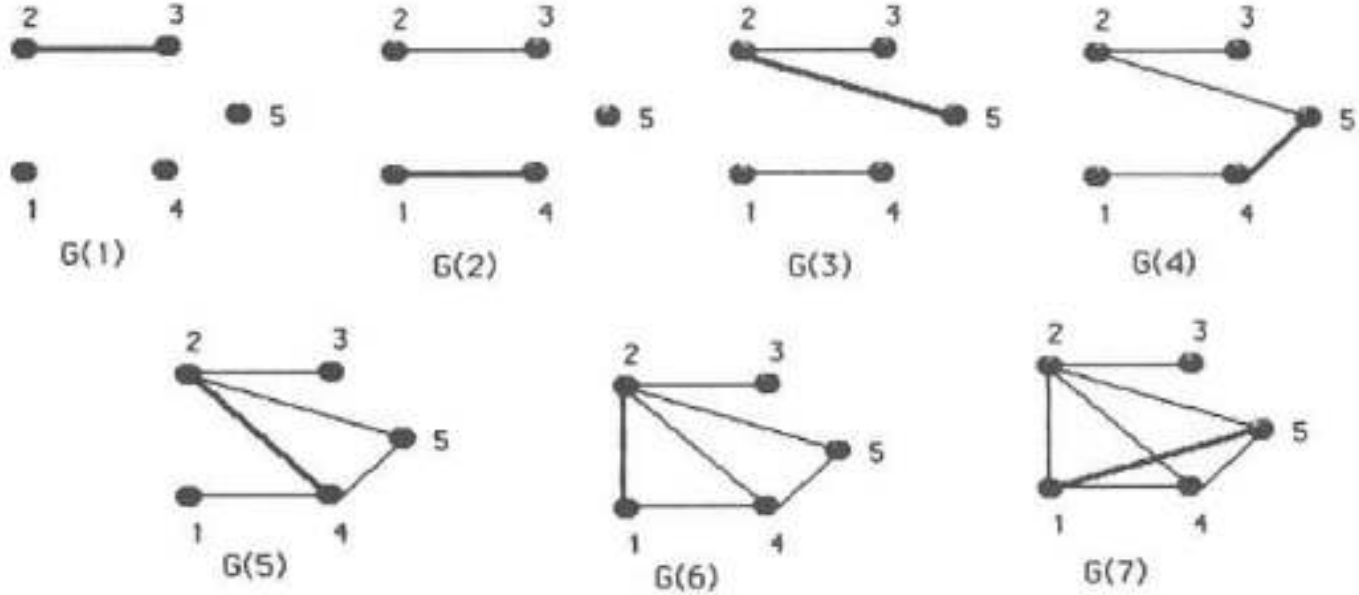
- Based on the equivalence between the thresholding of $|S|$ to the level λ and single-linkage clustering (Mirkin 1996, Jain & Dubes 1988)

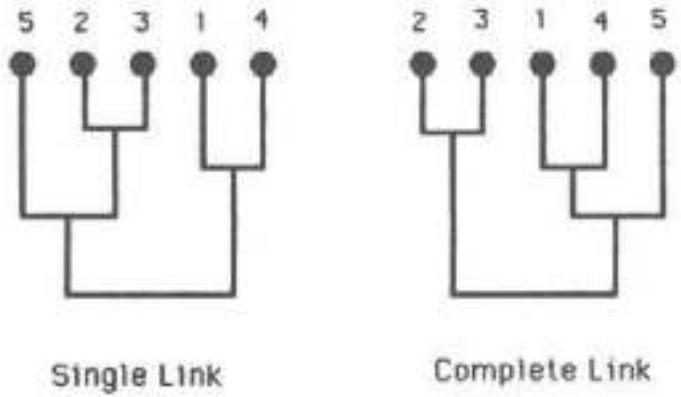
$$\mathscr{D}_1 = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} 0 & 6 & 8 & 2 & 7 \\ 6 & 0 & 1 & 5 & 3 \\ 8 & 1 & 0 & 10 & 9 \\ 2 & 5 & 10 & 0 & 4 \\ 7 & 3 & 9 & 4 & 0 \end{bmatrix} \end{array}$$





Links between clustering, cliques and connected components





Figures from Jain & Dubes 1988

Automatic selection of key genes prior to network inference

Cluster graphical lasso (Tan *et al.*, 2015) (inspired from the Block Diagonal Screening Rule, Mazumder et Hastie, 2012)

1. Detect K "blocks" of variables based on **average** linkage hierarchical clustering $\Rightarrow$ reduce the dimension of the network inference problem

2. Graphical lasso inference in each block with different regularization parameters

Automatic selection of key genes prior to network inference

Cluster graphical lasso (Tan *et al.*, 2015) (inspired from the Block Diagonal Screening Rule, Mazumder et Hastie, 2012)

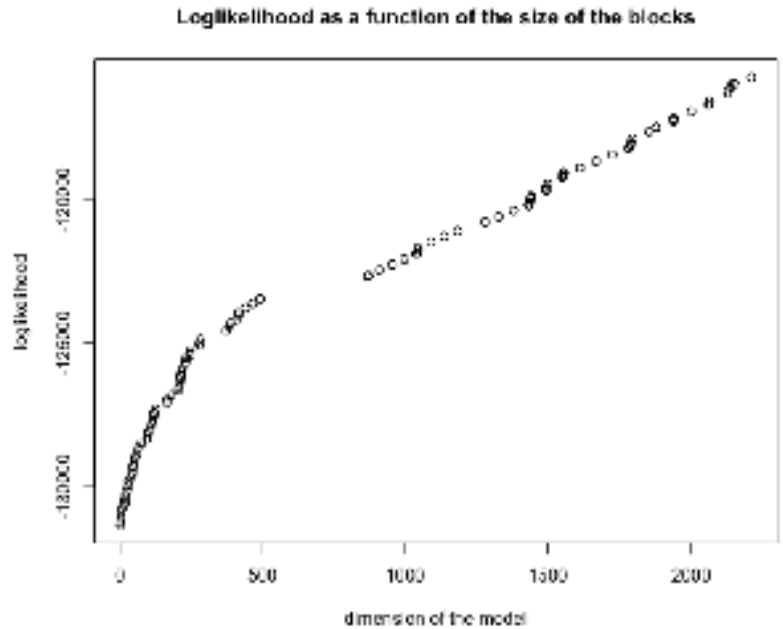
1. Detect K "blocks" of variables based on **average** linkage hierarchical clustering $\Rightarrow$ reduce the dimension of the network inference problem
   $\Rightarrow$ No clear rules to select K
2. Graphical lasso inference in each block with different regularization parameters

Automatic selection of key genes prior to network inference

Cluster graphical lasso (Tan *et al.*, 2015) (inspired from the Block Diagonal Screening Rule, Mazumder et Hastie, 2012)

1. Detect K "blocks" of variables based on **average** linkage hierarchical clustering $\Rightarrow$ reduce the dimension of the network inference problem
   $\Rightarrow$ No clear rules to select K
2. Graphical lasso inference in each block with different regularization parameters

How to select K? The following graph leads us to use the Slope heuristics



Model selection to detect groups of genes

$$\textbf{Hypothesis: } y_i \overset{\text{iid}}{\sim} \mathcal{N}_p(0, \Sigma_B) \text{ with } \Sigma_B = \begin{pmatrix} \Sigma^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma^K \end{pmatrix}$$

Model selection to detect groups of genes

Hypothesis: $\mathrm{y}_i \overset{\text{iid}}{\sim} \mathcal{N}_p(0, \Sigma_B)$ with $\Sigma_B = \begin{pmatrix} \Sigma^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma^K \end{pmatrix}$

The block structure of Σ_B provides a classification of genes $B = (B_1, \ldots, B_K)$

Model selection to detect groups of genes

$$\textbf{Hypothesis: } \mathsf{y}_i \overset{\text{iid}}{\sim} \mathcal{N}_p(0, \Sigma_B) \text{ with } \Sigma_B = \begin{pmatrix} \Sigma^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma^K \end{pmatrix}$$

The block structure of Σ_B provides a classification of genes $B = (B_1, \ldots, B_K)$

$$F_{\mathsf{B}} = \left\{ f_{\mathsf{B}} = \phi_p(0, \Sigma_{\mathsf{B}}) \text{ with } \Sigma_{\mathsf{B}} \in \mathbb{S}_p^{++}(\mathbb{R}) \;\middle|\; \Sigma_{\mathsf{B}} = P_\sigma \begin{pmatrix} \Sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma_K \end{pmatrix} P_\sigma^{-1}, \right\}$$

Model selection to detect groups of genes

$$\textbf{Hypothesis: } \mathsf{y}_i \overset{\text{iid}}{\sim} \mathcal{N}_p(0, \Sigma_B) \text{ with } \Sigma_B = \begin{pmatrix} \Sigma^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma^K \end{pmatrix}$$

The block structure of Σ_B provides a classification of genes $B = (B_1, \ldots, B_K)$

$$F_{\mathsf{B}} = \left\{ f_{\mathsf{B}} = \phi_p(0, \Sigma_{\mathsf{B}}) \text{ with } \Sigma_{\mathsf{B}} \in \mathbb{S}_p^{++}(\mathbb{R}) \;\middle|\; \begin{array}{l} \lambda_m \leq \Lambda_{\min}(\Sigma_{\mathsf{B}}) \leq \Lambda_{\max}(\Sigma_{\mathsf{B}}) \leq \lambda_M, \\ \Sigma_{\mathsf{B}} = P_\sigma \begin{pmatrix} \Sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma_K \end{pmatrix} P_\sigma^{-1}, \end{array} \right\}$$

Model selection to detect groups of genes

$$\textbf{Hypothesis: } \mathrm{y}_i \overset{\text{iid}}{\sim} \mathcal{N}_p(0, \Sigma_B) \text{ with } \Sigma_B = \begin{pmatrix} \Sigma^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma^K \end{pmatrix}$$

The block structure of Σ_B provides a classification of genes $B = (B_1, \ldots, B_K)$

$$F_{\mathrm{B}} = \left\{ f_{\mathrm{B}} = \phi_p(0, \Sigma_{\mathrm{B}}) \text{ with } \Sigma_{\mathrm{B}} \in \mathbb{S}_p^{++}(\mathbb{R}) \;\middle|\; \begin{array}{l} \lambda_m \leq \Lambda_{\min}(\Sigma_{\mathrm{B}}) \leq \Lambda_{\max}(\Sigma_{\mathrm{B}}) \leq \lambda_M, \\ \Sigma_{\mathrm{B}} = P_\sigma \begin{pmatrix} \Sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma_K \end{pmatrix} P_\sigma^{-1}, \end{array} \right\}$$

Selection of groups of genes based on non-asymptotic argument (Massart, 2003)

$$\hat{B} = \underset{B}{\operatorname{argmin}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log(\hat{f}_B(\mathrm{y}_i)) + \mathrm{pen}(B) \right\},$$

with $\mathrm{pen}(B)$ to define.

Selection among an aleatory sub-collection of models

$\mathcal{B}$: set of all possible partitions of the p variables

$\Rightarrow$ Exhaustive exploration of $\mathcal{B}$ is unrealistic

$\mathcal{B}^\wedge$: set of partitions obtained by thresholding of $|\, S \,|$

$$\hat{B} = \underset{B \in \mathcal{B}^\wedge}{\operatorname{argmin}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log(\hat{f}_B(y_i)) + \operatorname{pen}(B) \right\},$$

with $\operatorname{pen}(B)$ to define.

Oracle inequality

There exists some absolute constants κ and C_{oracle} such that whenever

$$\text{pen}(\mathsf{B}) \geq \kappa \frac{D_{\mathsf{B}}}{n} \left[2c^2 + \log\left(\frac{p^4}{D_{\mathsf{B}}(\frac{D_{\mathbf{B}}}{n} c^2 \wedge 1)} \right) \right]$$

for every $\mathsf{B} \in \mathcal{B}$, with $c = \sqrt{\pi} + \sqrt{\log(3\sqrt{3}\frac{\lambda_M}{\lambda_m})}$, the random variable $\hat{\mathsf{B}} \in \mathcal{B}_\Lambda$ such that

$$\hat{\mathsf{B}} = \underset{\mathsf{B} \in \mathcal{B}_\Lambda}{\text{argmin}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log(\hat{f}_{\mathsf{B}}(\mathsf{y}_i)) + \text{pen}(\mathsf{B}) \right\}$$

exists and, moreover, whatever the true density $f^\star$,

$$\mathbb{E}(\mathsf{d}_H^2(f^\star, \hat{f}_{\hat{\mathsf{B}}})) \leq C_{\text{oracle}} \mathbb{E}\left[\inf_{\mathsf{B} \in \mathcal{B}_\Lambda} \left(\inf_{f \in F_{\mathbf{B}}} \mathsf{KL}(f^\star, f) + \text{pen}(\mathsf{B}) \right) \right] + \frac{1 \vee \tau}{n} p \log(p).$$

Oracle inequality

There exists some absolute constants κ and C_{oracle} such that whenever

$$\text{pen}(\mathsf{B}) \geq \kappa \frac{D_{\mathsf{B}}}{n} \left[2c^2 + \log\left(\frac{p^4}{D_{\mathsf{B}}(\frac{D_{\mathbf{B}}}{n} c^2 \wedge 1)} \right) \right]$$

for every $\mathsf{B} \in \mathcal{B}$, with $c = \sqrt{\pi} + \sqrt{\log(3\sqrt{3}\frac{\lambda_M}{\lambda_m})}$, the random variable $\hat{\mathsf{B}} \in \mathcal{B}_\Lambda$ such that

$$\hat{\mathsf{B}} = \underset{\mathsf{B} \in \mathcal{B}_\Lambda}{\text{argmin}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log(\hat{f}_{\mathsf{B}}(\mathsf{y}_i)) + \text{pen}(\mathsf{B}) \right\}$$

exists and, moreover, whatever the true density $f^\star$,

$$\mathbb{E}(\mathsf{d}_H^2(f^\star, \hat{f}_{\hat{\mathsf{B}}})) \leq C_{\text{oracle}} \mathbb{E}\left[\inf_{\mathsf{B} \in \mathcal{B}_\Lambda} \left(\inf_{f \in F_{\mathbf{B}}} \mathsf{KL}(f^\star, f) + \text{pen}(\mathsf{B}) \right) \right] + \frac{1 \vee \tau}{n} p \log(p).$$

Minimax lower bound

Let $\mathsf{B} \in \mathcal{B}$. Consider the model F_{B} and D_{B} its dimension. Then, if we denote $C_{\text{minim}} = \frac{e}{4(2e+1)^2(8+\log(\lambda_M/\lambda_m))}$, for any estimator $\hat{f}_{\mathsf{B}}$ of $f^\star$ one has

$$\sup_{f^\star \in F_{\mathbf{B}}} \mathbb{E}(\mathsf{d}_H^2(\hat{f}_{\mathsf{B}}, f^\star)) \geq C_{\text{minim}} \frac{D_{\mathsf{B}}}{n} \left(1 + \log\left(\frac{2\lambda_M p(p-1)}{D_{\mathsf{B}}} \right) \right).$$

Our procedure

Step A (Block-diagonal covariance structure detection) Select the modularity structure of the network.

Our procedure

Step A (Block-diagonal covariance structure detection) Select the modularity structure of the network.

(1) Compute the sample covariance matrix S from the data y.

Our procedure

Step A (Block-diagonal covariance structure detection) Select the modularity structure of the network.

(1) Compute the sample covariance matrix S from the data y.

(2) Construct the sub-collection of partitions $\mathcal{B}_\Lambda = (\mathrm{B}_\lambda)_{\lambda \in \Lambda}$ by thresholding S, where Λ is a set of thresholds

Our procedure

Step A (Block-diagonal covariance structure detection) Select the modularity structure of the network.

(1) Compute the sample covariance matrix S from the data y.

(2) Construct the sub-collection of partitions $\mathcal{B}_\Lambda = (\mathrm{B}_\lambda)_{\lambda \in \Lambda}$ by thresholding S, where Λ is a set of thresholds

(3) For each partition $\mathrm{B} \in \mathcal{B}_\Lambda$, compute the corresponding maximum log-likelihood of the model.

Our procedure

Step A (Block-diagonal covariance structure detection) Select the modularity structure of the network.

(1) Compute the sample covariance matrix S from the data y.

(2) Construct the sub-collection of partitions $\mathcal{B}_\Lambda = (\mathrm{B}_\lambda)_{\lambda \in \Lambda}$ by thresholding S, where Λ is a set of thresholds

(3) For each partition $\mathrm{B} \in \mathcal{B}_\Lambda$, compute the corresponding maximum log-likelihood of the model.

(4) Based on the log-likelihood associated to each partition B in $\mathcal{B}_\Lambda$, **select $\hat{B}$ based on model selection**

Our procedure

Step A (Block-diagonal covariance structure detection) Select the modularity structure of the network.

(1) Compute the sample covariance matrix S from the data y.

(2) Construct the sub-collection of partitions $\mathcal{B}_\Lambda = (\mathrm{B}_\lambda)_{\lambda \in \Lambda}$ by thresholding S, where Λ is a set of thresholds

(3) For each partition $\mathrm{B} \in \mathcal{B}_\Lambda$, compute the corresponding maximum log-likelihood of the model.

(4) Based on the log-likelihood associated to each partition B in $\mathcal{B}_\Lambda$, **select $\hat{B}$ based on model selection**

Step B (Network inference in each module) For each group of variables in the selected partition $\hat{\mathrm{B}}$, infer the network using the graphical lasso.

Our procedure

Step A (Block-diagonal covariance structure detection) Select the modularity structure of the network.

(1) Compute the sample covariance matrix S from the data y.

(2) Construct the sub-collection of partitions $\mathcal{B}_\Lambda = (\mathrm{B}_\lambda)_{\lambda \in \Lambda}$ by thresholding S, where Λ is a set of thresholds

(3) For each partition $\mathrm{B} \in \mathcal{B}_\Lambda$, compute the corresponding maximum log-likelihood of the model.

(4) Based on the log-likelihood associated to each partition B in $\mathcal{B}_\Lambda$, **select $\hat{B}$ based on model selection**
$\Rightarrow$ Non-asymptotic theoretical guaranties for model selection

Step B (Network inference in each module) For each group of variables in the selected partition $\hat{\mathrm{B}}$, infer the network using the graphical lasso.

Oracle inequality

There exists some absolute constants κ and C_{oracle} such that whenever

$$\text{pen}(\mathsf{B}) \geq \kappa \frac{D_{\mathsf{B}}}{n} \left[2c^2 + \log\left(\frac{p^4}{D_{\mathsf{B}}(\frac{D_{\mathsf{B}}}{n} c^2 \wedge 1)} \right) \right]$$

for every $\mathsf{B} \in \mathcal{B}$, with $c = \sqrt{\pi} + \sqrt{\log(3\sqrt{3}\frac{\lambda_M}{\lambda_m})}$, the random variable $\hat{\mathsf{B}} \in \mathcal{B}_\Lambda$ such that

$$\hat{\mathsf{B}} = \underset{\mathsf{B} \in \mathcal{B}_\Lambda}{\text{argmin}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log(\hat{f}_{\mathsf{B}}(\mathsf{y}_i)) + \text{pen}(\mathsf{B}) \right\}$$

exists and, moreover, whatever the true density $f^\star$,

$$\mathbb{E}(\mathsf{d}_H^2(f^\star, \hat{f}_{\hat{\mathsf{B}}})) \leq C_{\text{oracle}} \mathbb{E}\left[\inf_{\mathsf{B} \in \mathcal{B}_\Lambda} \left(\inf_{f \in F_{\mathsf{B}}} \text{KL}(f^\star, f) + \text{pen}(\mathsf{B}) \right) \right] + \frac{1 \vee \tau}{n} p \log(p).$$

Minimax lower bound

Let $\mathsf{B} \in \mathcal{B}$. Consider the model F_{B} and D_{B} its dimension. Then, if we denote $C_{\text{minim}} = \frac{e}{4(2e+1)^2(8+\log(\lambda_M/\lambda_m))}$, for any estimator $\hat{f}_{\mathsf{B}}$ of $f^\star$ one has

$$\sup_{f^\star \in F_{\mathsf{B}}} \mathbb{E}(\mathsf{d}_H^2(\hat{f}_{\mathsf{B}}, f^\star)) \geq C_{\text{minim}} \frac{D_{\mathsf{B}}}{n} \left(1 + \log\left(\frac{2\lambda_M p(p-1)}{D_{\mathsf{B}}} \right) \right).$$

Our procedure

Shock procedure : **S**lope **h**euristic for bl**ock**-diagonal covariance structure detection for network inference

Step A (Block-diagonal covariance structure detection) Select the modularity structure of the network.

(1) Compute the sample covariance matrix S from the data y.

(2) Construct the sub-collection of partitions $\mathcal{B}_\Lambda = (\mathrm{B}_\lambda)_{\lambda \in \Lambda}$ by thresholding S, where Λ is a set of thresholds

(3) For each partition $\mathrm{B} \in \mathcal{B}_\Lambda$, compute the corresponding maximum log-likelihood of the model.

(4) Based on the log-likelihood associated to each partition B in $\mathcal{B}_\Lambda$, **select $\hat{B}$ based on model selection**
$\Rightarrow$ Non-asymptotic theoretical guaranties for model selection

Step B (Network inference in each module) For each group of variables in the selected partition $\hat{\mathrm{B}}$, infer the network using the graphical lasso.

Our procedure

Shock procedure : **S**lope **h**euristic for bl**ock**-diagonal covariance structure detection for network inference

Step A (Block-diagonal covariance structure detection) Select the modularity structure of the network.

(1) Compute the sample covariance matrix S from the data y.

(2) Construct the sub-collection of partitions $\mathcal{B}_\Lambda = (\mathrm{B}_\lambda)_{\lambda \in \Lambda}$ by thresholding S, where Λ is a set of thresholds

(3) For each partition $\mathrm{B} \in \mathcal{B}_\Lambda$, compute the corresponding maximum log-likelihood of the model.

(4) Based on the log-likelihood associated to each partition B in $\mathcal{B}_\Lambda$, **select $\hat{B}$ based on model selection**
$\Rightarrow$ Non-asymptotic theoretical guaranties for model selection
$\Rightarrow$ In practice : we don't use the *theoretical* penalty, we calibrate the constant κ from the data using **the slope heuristic**

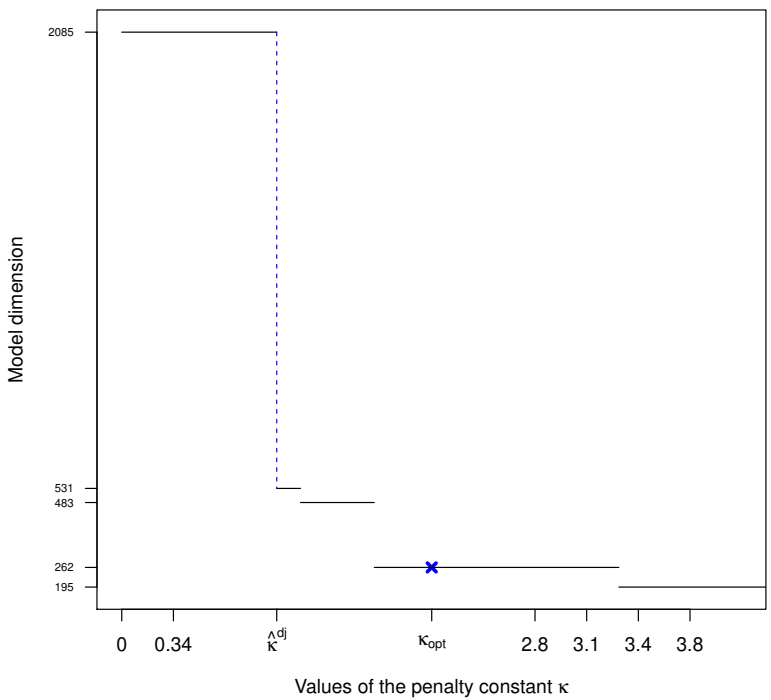
$$\mathrm{pen}(B) = \kappa D_B$$

Step B (Network inference in each module) For each group of variables in the selected partition $\hat{\mathrm{B}}$, infer the network using the graphical lasso.

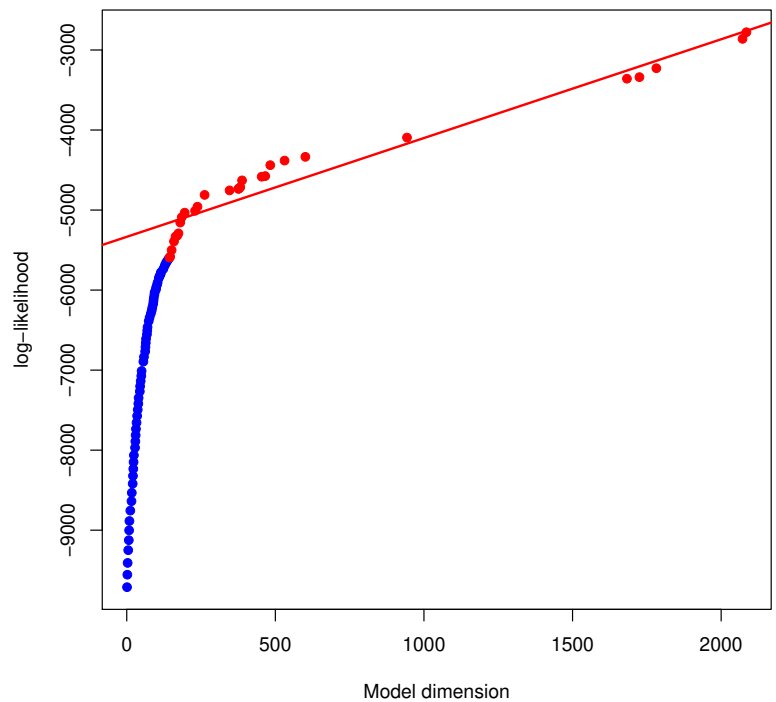
Calibration of coefficient κ in $\mathrm{pen}(B) = \kappa D_B$

- Illustrations on simulated data: $p = 100$, $n = 70$ et $K^\star = 15$
- Practical solution to calibrate the penalty
  implemented in the package R `capushe` (Baudry *et al.*, 2012)

Method 1: SHDJ
Slope Heuristics Dimension Jump

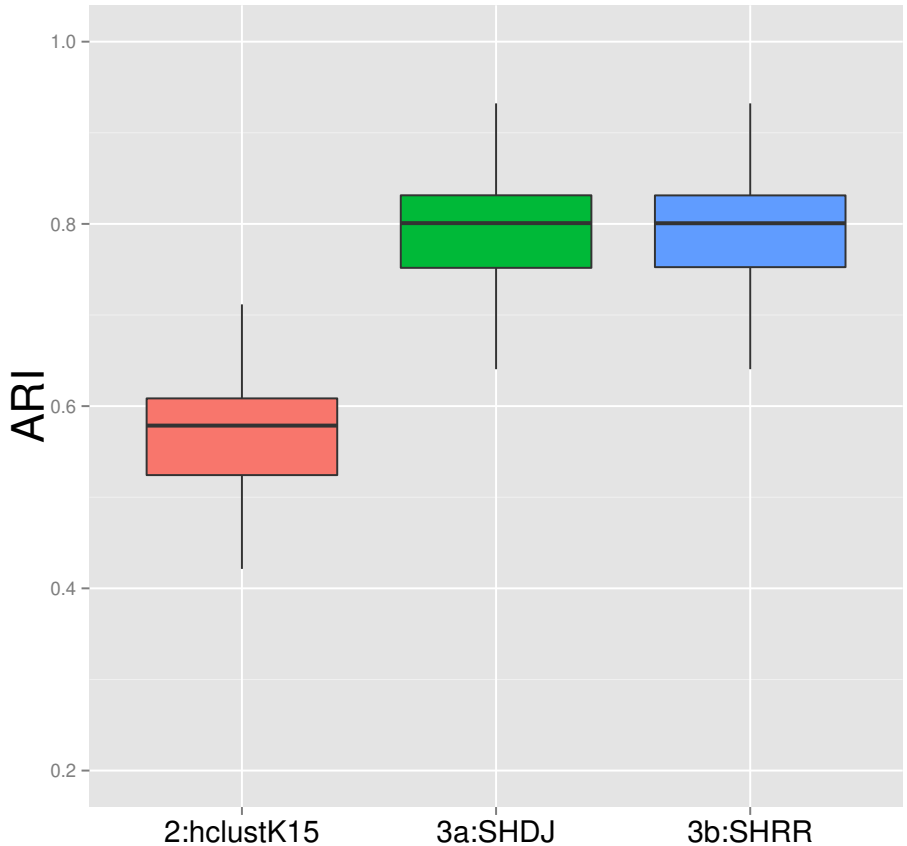


Method 2: SHRR
Slope Heuristics Robust Regression



Results on 100 replicated datasets

Simulated data: $p = 100$, $n = 70$ and Σ block diagonal with $K^\star = 15$.



Adjusted Rand Index

between the true partition and the selected partition

Comparison of four strategies to infer networks

Simulated data: $p = 100$, $n = 70$ and Σ block diagonal with $K^\star = 15$.

- **(1) Graphical lasso**
  Network inference on all variables (graphical lasso with BIC)

- **(2) Cluster Graphical Lasso** (Tan *et al.*, 2015)
  - Step 1: Hierarchical classification of variables, for fixed $K = K^\star$
  - Step 2: Graphical lasso with regularization parameters $\rho_1, \ldots, \rho_{K^\star}$ from Tan 2015.

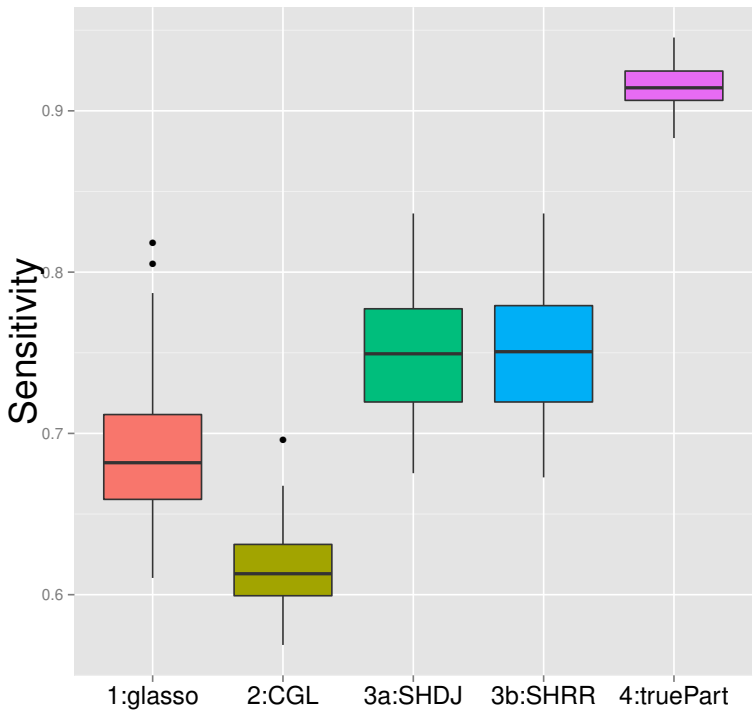
- **(3) Our solution**
  - Step 1: Non-asymptotic model selection of groups of genes
    - (3a) SHRR partition
    - (3b) SHDJ partition
  - Step 2: Network inference in each group (graphical lasso with BIC)

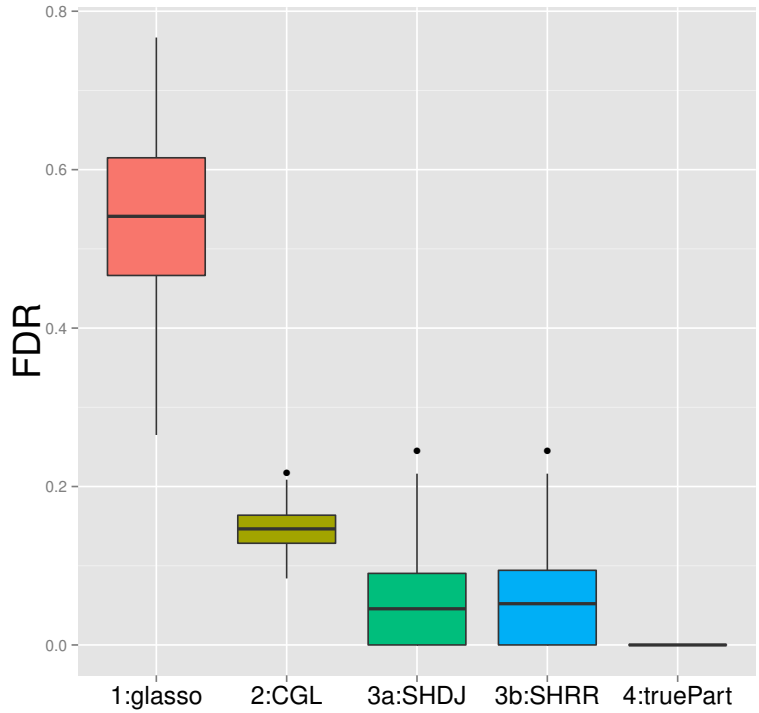
- **(4) True Partition**
  - Step 1: Set the partition of variables to the true partition (known)
  - Step 2: Network inference in each group (graphical lasso with BIC)

Performance of strategies in simulated data

Simulated data: $p = 100$, $n = 70$ and Σ block diagonal with $K^\star = 15$.

$$\text{Sensitivity} = \frac{TP}{(TP + FN)} \qquad FDR = \frac{FP}{(TP + FP)}$$

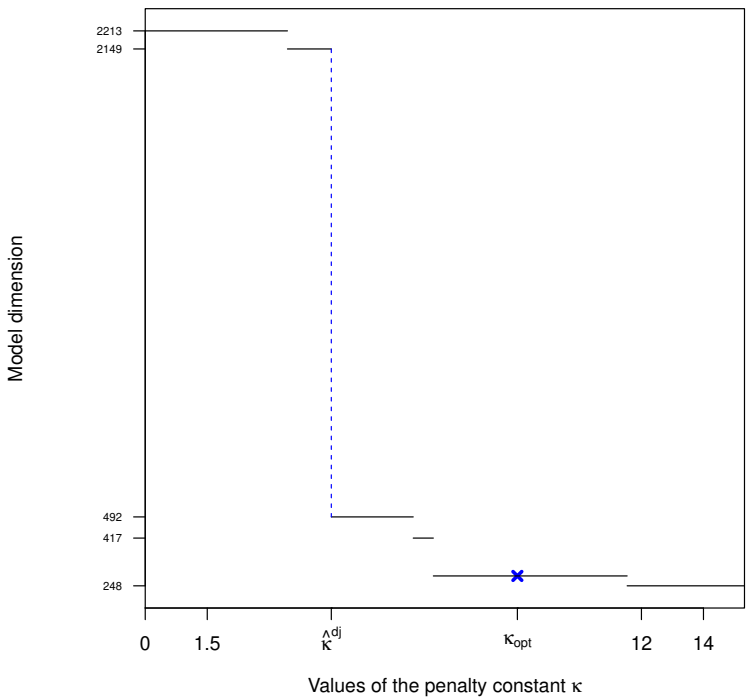


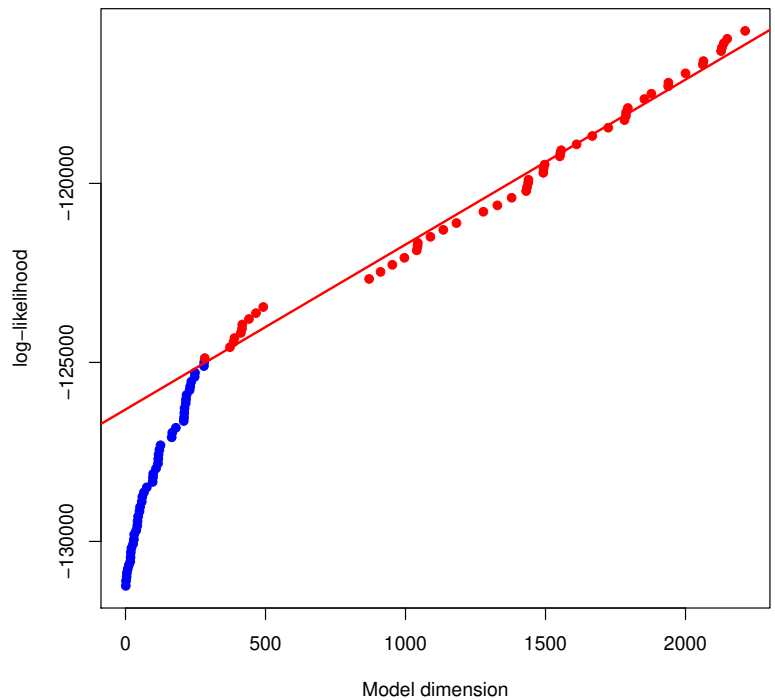


Results on 100 replicated datasets

Results on real data

- Pickrell *et al.* (2010): RNA sequencing from lymphoblastoid cell lines derived from $n = 69$ unrelated Nigerian individuals
- Selection of $p = 200$ highest variable genes





→ Partitions selected with SHRR and SHDJ are the same

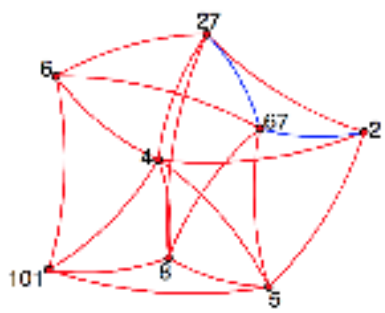
Network inference on Pickrell data

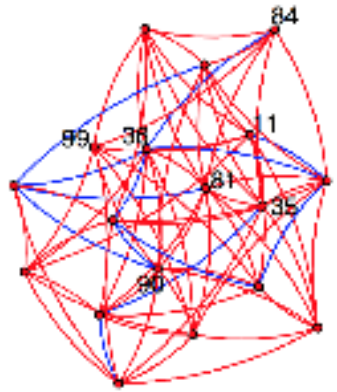
Graphical lasso

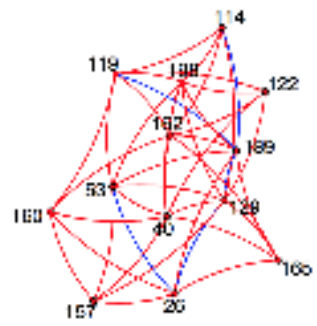
- $D = 19900$ parameters to estimate

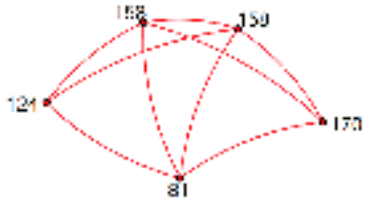
Partition detected by slope heuristic $\hat{B}$

- $D_{\hat{B}_{\mathbf{SH}}} = 283$ parameters to estimate
- $\hat{K}_{\mathrm{SH}} = 150$ blocks
- 140 blocks of size 1, 2 blocks of size 2, 4 blocks of size 3 and 4 blocks of size $18, 13, 8$ et 5









Bibliography

- J. Friedman, T. Hastie, R. Tibshirani, *Sparse inverse covariance estimation with the Lasso*, Biostatistics, 2008
- R. Mazumder, T. Hastie, *Exact Covariance Thresholding into Connected Components for Large-Scale Graphical Lasso*, Journal of Machine Learning Reasearch, 2012
- K. Tan, D. Witten, A. Shojaie, *The Cluster Graphical Lasso for improved estimation of Gaussian graphical models*, Computation Statistics & Data Analysis, 2015
- L. Birgé, P. Massart, *Minimal penalties for Gaussian model selection*, Probab. Theory Related Fields, 2007
- J.P. Baudry, C. Maugis, B. Michel, *Slope heuristics: overview and implementation*, Statistics and Computing, 2012
- E. Devijver, M. Gallopin, *Block-diagonal covariance selection for high-dimensional Gaussian graphical models*, Journal of the American Statistical Association, 2018.

Practical session

- Demonstration of the slope heuristic (SH) on gene expression :
  - Pickrell data : $n = 69$ individuals, $p = 200$ genes `geneExpression.RData`
  - `demoSHGeneExpression.R`
  - Requires `functions.R`
- To go further :
  - Network inference techniques are often criticized because they are known to be "unstable" : if we add more individuals to the dataset, the inferred network might change drastically.
  - **Question : is the partition of genes into subgroups detected by SH stable by resampling?**
  - Our hypothesis : the groups of genes detected by the slope heuristic has good "stability" properties!
  - A new dataset : `BRCA.RData` with $p = 200$ genes and more $n = 1212$ individuals (TCGA database). Are the partitions detected by SH on small subsamples ($n_{\text{sub}} = 70$) similar between each others?