

# Network Inference

[julien.chiquet@gmail.com](mailto:julien.chiquet@gmail.com), [guillem.rigaille@gmail.com](mailto:guillem.rigaille@gmail.com)

Summer School 2016 - From gene expression to genomic network

---

This practical aims to provide a quick overview of sparse Gaussian Graphical Models (GGM) and their use in the context of network reconstruction for gene interaction networks.

To this end, we rely on the R-package **huge**, which implements some of the most popular sparse GGM methods and provides a set of basic tools for their handling and their analysis.

The first part focuses on an empirical analysis of the statistical models used for network reconstruction. The objective is to quickly study the range of applicability of these methods. It should also give you some insights about their limitations, especially toward the interpretability of the inferred network in terms of biology.

The second part applies these methods to two data sets: the first one consists in a transcriptomic data associated to a small regulatory network (tens of genes) known by the biologists. The second one is a large cohort of breast cancer transcriptomic data set associated to 44,000 transcripts. The objective is to unravel the most striking interactions between differentially expressed genes.

*Note* : you can form small and balanced groups of students to work. Some function required during the session are available in file `external_functions.R` (ask the teachers).

---

## 1 First part: empirical study of sparse GGM

Load the **huge** package. Have a quick glance at the help.

### 1.1 Synthetic data generation, Network representation

The function `huge.generator` allows to generate a random network and some (expression) data associated with this network.

- Use this function to generate a simple random network with the size of your choice. Have a look at the structure of the R object produced by the function.
- Try different network typologies and plot the outputs with the dedicated `plot` function. Comment and explain what represent the different graphical outputs.
- Have a look at the distribution of the data generated. You may use `hist`, `density`, `qqnorm` and so on. Comment.
- How can you change the level of difficulty in the problem of network reconstruction?

## 1.2 Correlation vs. Partial correlation

This section aims to illustrate the difference of relationship modeled by *correlation* and *partial correlation*.

- generate a graph with  $d = 10$  nodes, a single hub, and expression data with  $n = 200$  samples. We are going to study the statistical relationship between the hub (first node in your graph) and 2 of its neighbors. We call `hub` the index of the hub and `neighbor1`, `neighbor2` the label (index) of these three nodes.
- Adjust three simple linear regressions between the data associated with these three nodes, that is, between `hub` and `neighbor1`, `hub` and `neighbor2`, and `neighbor1` and `neighbor2`. Use the function `lm + summary` to test the significance of each model. Comment.
- Partial correlation corresponds to correlation that remains between pairs of variables once remove the effect of the others variables. To test direct relationships between two neighbors, with thus have to remove the effect of all others variables. To do so, adjust two multiple linear model to predict the expression of the two neighbors by all the others genes but them.
- Get back the residuals associated with each model. This corresponds to what is not predicted in the expression of each neighbor by all the genes but `neighbor1` and `neighbor2`.
- Adjust simple linear regression between the residuals and test the significance of this model. Comment
- Use the `geom_smooth` function in the package `ggplot2` to adjust and represent the four simple linear regression models fitted so far and summary your experiment.

### 1.2.1 Network inference accuracy

Now, to the network reconstruction at last ! The function `huge` automatically select the most significant partial correlation between variables, by adjusting a sparse GGM to the data. The final number of interactions (i.e., the number of edge in the reconstructed network) is controlled by a tuning parameter, the choice of which can be obtained by cross-validation.

We want to study the effect of the sample size on the performance of two methods: the sparse GGM approach (hereafter `glasso`) and the simple correlation approach, which just consists in thresholding the matrix of empirical correlations.

- The following function `one.simu` performs a simulation by computing the area under ROC curve for the `glasso` and the `correlation` based approach from a data set generated with the `huge.generator` function. The number of genes is 25, and the sample size varies from 5 to 500. Read it, try to understand it...

```
require(reshape2)
one.simu <- function(i) {
  cat("n =")
  d <- 25; seq.n <- c(5,15,30,50,100,250,500)
  out <- data.frame(t(sapply(seq.n, function(n) {
    cat("",n)
    exp <- huge.generator(n, d, verbose=FALSE)
    glasso <- huge(exp$data, method="glasso", verbose=FALSE)
    corthr <- huge(exp$data, method="ct", verbose=FALSE)
    res.corthr <- perf.auc(perf.roc(corthr$path, exp$theta))
    res.glasso <- perf.auc(perf.roc(glasso$path, exp$theta))
    return(setNames(c(res.glasso,res.corthr,n,i),
                     c("glasso","correlation","sample size", "simu")))
  })))
  return(melt(out, measure.vars = 1:2, value.name = "score"))
}
```

- Use this function to perform one single simulation and represent the evolution of the AUC for each method as a function of the sample size.
- Perform a bunch (say 30) simulations and represent the boxplots of the AUC for each method and as a function of the sample size. You may use the `parallel` package with its function `mclapply`. (or `doMC` and `foreach` if you work with Windows). This might take some time, so check your code and be patient!

## 2 Second part: transcriptomic data analysis

### 2.1 Inferring a network from transcriptomic data, normalization?

Now we will have look at the gene expression levels of 16 genes part of the core network identified by Françoise Monéger and her colleagues. In total we have 20 measurements : 2 times 10 biological replicates of flower bud.

First, load the data and the network identified by Françoise Monéger and colleagues. (the network is store as 16 x 16 contingency matrix).

```
load( file="data_school/Expr.RData")
load(file="data_school/Netw_FM.RData")
```

Then load a few usefull functions

```
### Inference
source("external_functions.R")
require(huge)
### a simple function thresholding a matrix for nlambda values
### between the min and max value (outside of the diagonal)
cor.thres <- function(C.mat){
  diag(C.mat) <- 0
  C.mat <- abs(C.mat)
  lvls <- c(0, unique(C.mat), max(C.mat)+1)
  res <- lapply( lvls,
                 FUN=function(thres) ((C.mat >= thres)+0))
}
```

### 2.1.1 Inferring the network with sparse GGM or correlation network

Infer the network using correlation or glasso and draw the ROC of the two approaches.

*Importantly, for the glasso approach it might be necessary to manually set the grid of lambdas.*

### 2.1.2 Normalisation and network inference

The huge package provides various way to pre-process the data (and try to make them more “normal”). Try to use the “skeptical” approach (`huge.npn(X, npn.func="skeptical")`) and draw its ROC curve. What do you conclude ?

### 2.1.3 Inversion correlation ?

Given the number of samples we could also try to inverse the correlation matrix directly. Try this other approach and draw its ROC curve and conclude.

## 2.2 Differential analysis and gene networks

Now we will look at some data from Guedj et al. (2011). In this data there are two groups ER positive breast tumors and ER negative breast tumors. A number of genes are differentially expressed between these two groups.

We will first load the data

```
load ("data_school/breast_cancer_guedj11.RData") # raw data
load ("data_school/gen_name.RData") # each row of the raw data matrix is a gene.
gene.name <- unlist(gene.name)
data.raw <- expr
```

### 2.2.1 Differential analysis

Run limma or a t.test analysis to compare the two groups. Assess the number of significant genes (with an adjusted p-value below 0.05).

### 2.2.2 Network inference

Infer a network for each group using the first  $p=100$  most differentially expressed genes. Check the shape of the ebic to assess the quality of the inference and the grid of lambdas.

### 2.2.3 ESR1 edges in ER positive tumors

Identify ESR1 edges that are only present in ER positive tumors.

### 2.2.4 Stability of those edges

Assess whether those edges are stable using resampling. Search "FOXA1" in google scholar or entrez.

### 2.2.5 Correlation

Compute the correlation between ESR1 and the top 100 genes in ER positive and ER negative tumors. Look at FOXA1.