Inferring a biological network from transcriptomic data and using (some standard) GGMs approaches

Guillem Rigaill

October 12, 2016

Correlation, partial correlation and GGMs

Correlation and partial correlation in simulations

An application to a well defined biological problem

An application to a “less well” defined biological problem

Correlation, partial correlation and GGMs

Models and application to transcriptomic data

1. Some well defined statistical models, quantities and goals
   - Covariance
   - Inverse-covariance
   - Goal: Infer the network
     - non-zero coefficients of the inverse-covariance matrix
2. Some less well defined biological goals and questions
   - Sometimes?: infer the "whole network" This is difficult because:
     - We only have transcriptomic data ?
     - We don't look at DNA, RNA, protein, miRNA...
     - We have a mixed population of cells
     - We have "little" data $n << p$
     - We don't know which gene we should look at
   - More often?: detecting interesting (direct) interactions ?
     - key genes, key groups of genes...

Correlation to infer a network

- Compute the correlation matrix
  - Fairly easy $\mathcal{O}(n^2)$
  - Similar expression profiles $\rightarrow$ high-correlation
    - co-regulation ?
  - Predict an edge between two genes if their absolute correlation is above a given threshold
  - Some questions/difficulties:
    - How to set the threshold ?
    - If we aim at recovering genes with similar expression profiles why not clustering ?
    - If we have enough data we can do better than that and detect direct interactions using partial correlation and GGMs

Graphical model

Definition: A graphical model gives a graphical (intuitive) representation of the dependence structure of a probability distribution. It links:

- a random vector $X = \{X_1, \ldots, X_p\}$ with distribution $\mathbb{P}$,
- a graph $\mathcal{G} = (\mathcal{P}, \mathcal{E})$ where
  - $\mathcal{P} = \{1, \ldots, p\}$ is the set of nodes associated to each variable,
  - $\mathcal{E}$ is a set of edges describing the dependence relationship of $X \sim \mathbb{P}$.

Conditional independence graph: It is the undirected graph $\mathcal{G} = \{\mathcal{P}, \mathcal{E}\}$ where

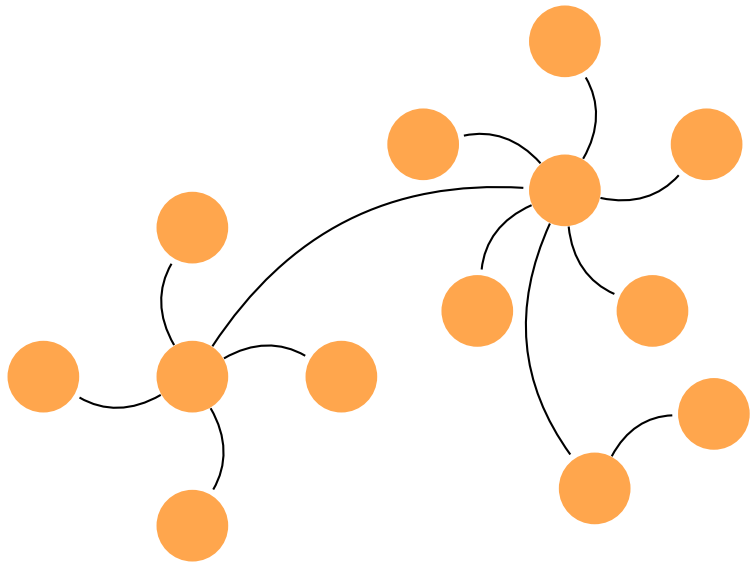
$$(i,j) \notin \mathcal{E} \Leftrightarrow X_i \perp\!\!\!\perp X_j | \mathcal{P} \backslash \{i,j\}.$$

Gaussian Graphical models

Multivariate Gaussian assumption Let $\mathbf{X} \sim \mathcal{N}(\mathbf{0}_p, \mathbf{\Sigma})$ and $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ the precision matrix.

$$\mathbf{x}^{(1)} = (x_1^{(1)}, ..., x_p^{(1)})$$
$$\mathbf{x}^{(2)} = (x_1^{(2)}, ..., x_p^{(2)})$$
$$...$$
$$\mathbf{x}^{(n)} = (x_1^{(n)}, ..., x_p^{(n)})$$

Inference



i.i.d. sample

$\mathcal{G} = (\mathcal{P}, \mathcal{E})$

1

[1] kindly provided by J. Chiquet

GGM and partial covariance

1. Suppose $X \sim \mathcal{N}(\boldsymbol{\mu}, \begin{pmatrix} \Sigma_{aa} & \Sigma_{ba} \\ \Sigma_{ab} & \Sigma_{bb} \end{pmatrix})$, then
   - X_a is Gaussian with distribution $\mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})$
   - $X_a | X_b = x$ is Gaussian with distribution $\mathcal{N}(\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b})$.
2. Partial covariance/correlation and conditional independance
   - Let X, Y, Z be real random variables.

$$\mathrm{cov}(X, Y|Z) = \mathrm{cov}(X, Y) - \mathrm{cov}(X, Z)\mathrm{cov}(Y, Z)/\mathrm{Var}(Z).$$

   - When X, Y, Z are jointly Gaussian, then

$$\mathrm{cov}(X, Y|Z) = 0 \Leftrightarrow \mathrm{cor}(X, Y|Z) = 0 \Leftrightarrow X \perp\!\!\!\perp Y|Z.$$

Gold standard convexified penalized approaches

1. Penalized likelihood (Banerjee et al., Yuan and Lin, 2008)}

$$\hat{\mathbf{\Theta}}_\lambda = \arg\max_{\mathbf{\Theta}\in\mathbb{S}_+} \ell(\mathbf{\Theta};\mathbf{X}) - \lambda\|\mathbf{\Theta}\|_1$$

2. Neighborhood Selection (Meinshausen and Bülhman, 2006)}

$$\widehat{\beta}^{(i)} = \arg\min_{\beta\in\mathbb{R}^{p-1}} \frac{1}{n}\left\|\mathbf{X}_i - \mathbf{X}_{\setminus i}\,\beta\right\|_2^2 + \lambda\,\|\beta\|_1$$

3. CLIME – Pseudo-likelihood (Cai et al., 2011; Yuan, 2010)}

$$\widehat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} \|\mathbf{\Theta}\|_1 \text{ subjected to } \left\|n^{-1}\mathbf{X}^t\mathbf{X}\mathbf{\Theta} - \mathbf{I}\right\|_\infty \leq \lambda$$

Some theoretical results with practical applications

1. **Selection consistency** (Ravikumar, Wainwright, 2009-2012) Denote $d = \max_{j \in \mathcal{P}}(\mathrm{degree_j})$. Consistency for an appropriate λ and
   - $n \approx \mathcal{O}(d^2 \log(p))$ for the graphical Lasso and Clime.
   - $n \approx \mathcal{O}(d \log(p))$ for neighborhood selection (sharp).
2. **Ultra high-dimension phenomenon** (Verzelen, 2011)

Minimax risk for sparse regression with d-sparse models blows-up when

$$\frac{d \log(p/d)}{n} \geq 1/2, \qquad (\mathrm{e.g.}, n = 50, p = 200, d \geq 8).$$

The R package huge

Here I will use the R package **huge** (Zhao et al. 2012):

- fairly easy to use
- semi-parametric version (copula...)
- some model selection tools (ebic, stars)
- some simulation fonctions

```
suppressMessages(
  library(huge, quietly=TRUE)
  )
```

Some datasets to test some standard GGMs approaches ?

1. Simulated data
   - Test that an approach is working under some simple conditions
   - Especially usefull when the approach has no underlying model
   - Essential sanity check
2. Arabidopsis thaliana data (infer a network)
   - 10 samples (sepal primordium)
   - 35 genes involved in a known network (F. Monéger)
   - Goal: can we recover part of the network using GGMs ?
3. Breast cancer data (pinpoint interesting genes/pathways)
   - Several hundred breast cancers (estrogen receptor + and -)
   - Several thousand genes
   - Goal: How can GGMs approaches help ?

Correlation and partial correlation in simulations

Some simple simulations (network with hubs)

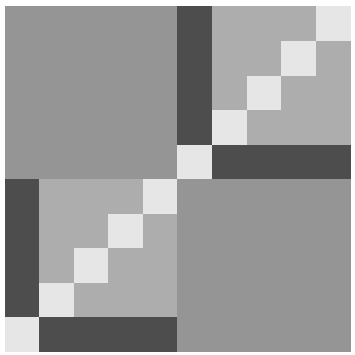
```
set.seed(11)
n <- 80; d <- 10;
rd.net  <- huge.generator(n, ## number of samples
                          d, ## number of genes
                          graph="hub", ## type of net
                          g=2, ## number of group)
                          verbose=FALSE)
```

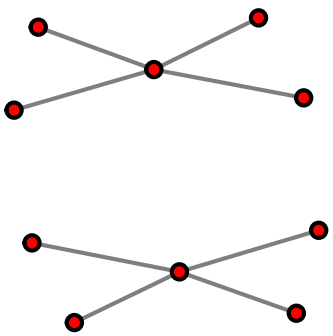
Some simple simulations (network with hubs)

```
plot(rd.net)
```

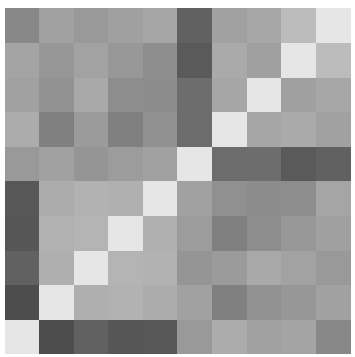
Adjacency Matrix

Covariance Matrix





Empirical Covariance Matrix



Inference using GGMs and correlation

1. Inference

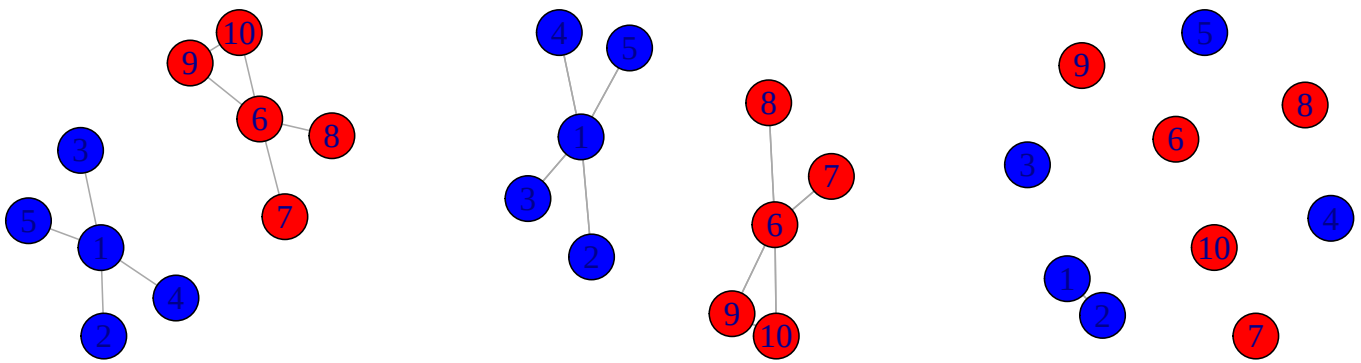
```
## glasso, mb and ct
glasso <- huge(rd.net$data, method="glasso",
               nlambda=50,  verbose=F)
mb <- huge(rd.net$data, method="mb",
           nlambda=50, verbose=F)
corthr <- huge(rd.net$data, method="ct",
              nlambda = 50, verbose=F)
```

2. Selection

```
## glasso, mb and ct
glasso.sel <- huge.select(glasso, "stars", verbose=F)
mb.sel <- huge.select(mb, "stars", verbose=F)
corthr.sel <- huge.select(corthr, "stars", verbose=F)
```

Inference using GGMs and correlation (results)

```
gr.glasso <- graph.adjacency(glasso.sel$refit)
V(gr.glasso)$label.cex <- 2
V(gr.glasso)$color <- rep(c("blue", "red"), each=5)
par(mfrow=c(1, 3))
plot(gr.glasso, vertex.size=30, edge.arrow.mode = "-")
plot(gr.mb, vertex.size=30, edge.arrow.mode = "-")
plot(gr.cor, vertex.size=30, edge.arrow.mode = "-")
```



A bit of code to run a simulation

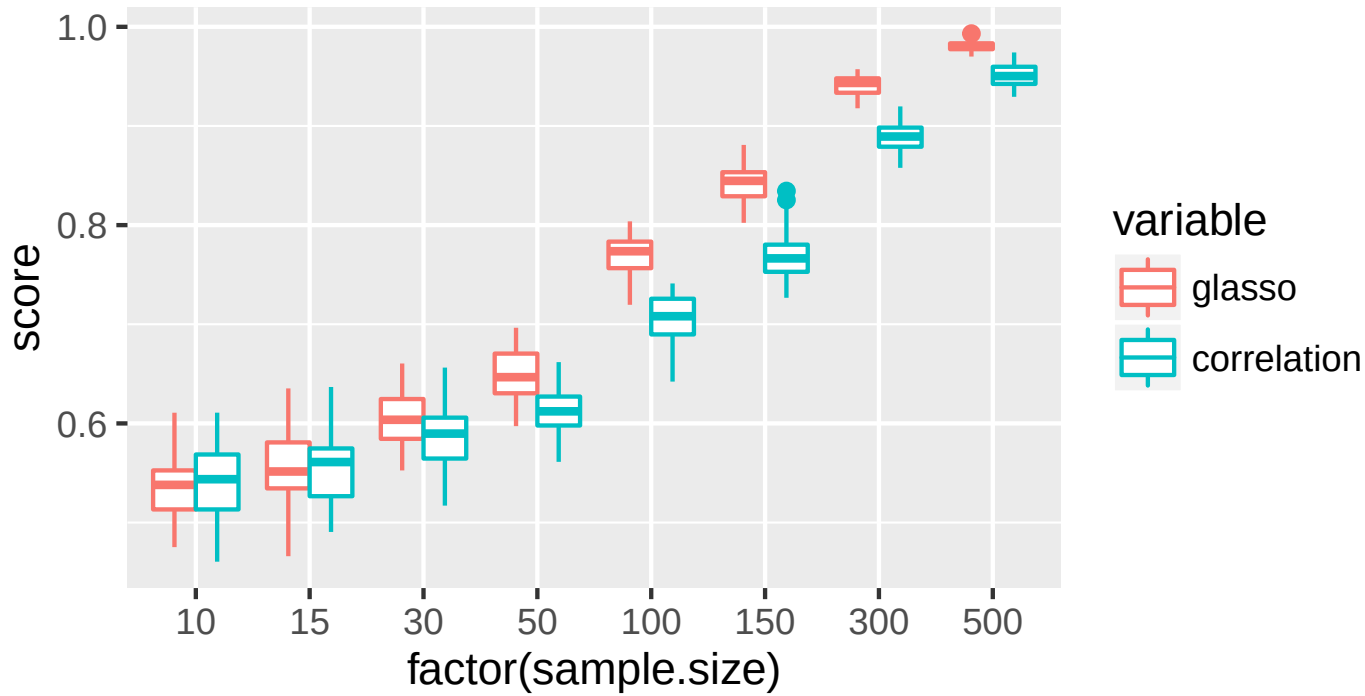
```
suppressMessages(require(reshape2))
one.simu <- function(i) {
  lbd.c <- seq(1, 0, -10^-2);
  d <- 25; seq.n <- c(10, 15, 30, 50, 100, 150, 300, 500)
  out <- data.frame(t(sapply(seq.n, function(n) {
   exp <- huge.generator(n, d, graph="cluster",
                         g=3, prob=1, verbose=F)
   gl <- huge(exp$data, method="glasso", nlambda=50, verbos
   cthr <- huge(exp$data, method="ct", lambda=lbd.c, verbos
   res.cthr <- perf.auc(perf.roc(cthr$path, exp$theta))
   res.gl <- perf.auc(perf.roc(gl$path, exp$theta))
   return(setNames(c(res.gl,res.cthr,n,i),
   c("glasso","correlation","sample size", "simu")))
  })))
return(melt(out, measure.vars = 1:2, value.name = "score"))
```

Run

```
suppressMessages(library(parallel))
res <- do.call(rbind, mclapply(1:40, one.simu, mc.cores=4))
```

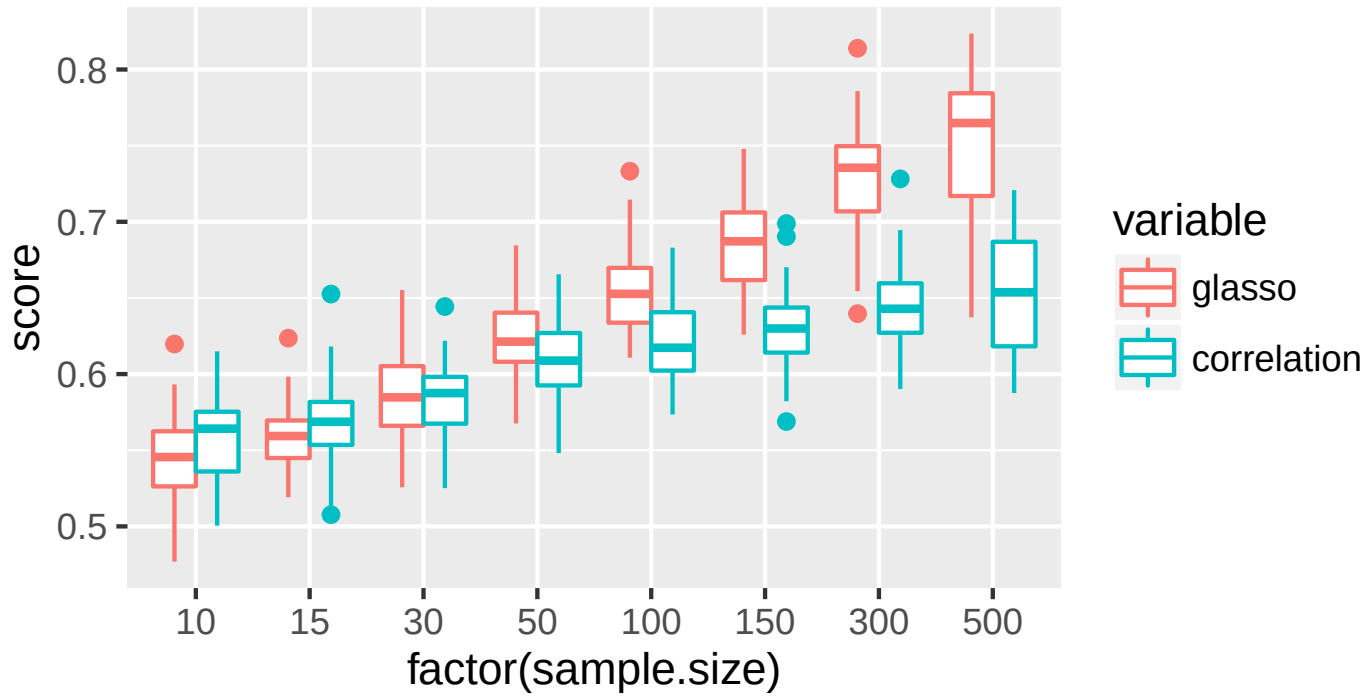
Simulation results (cluster - clique)

```
suppressMessages(library(ggplot2))
ggplot(res, aes(x=factor(sample.size), y=score)) +
#geom_point(alpha=.5, aes(group=variable,colour=variable))
geom_boxplot(aes(colour=variable))
```



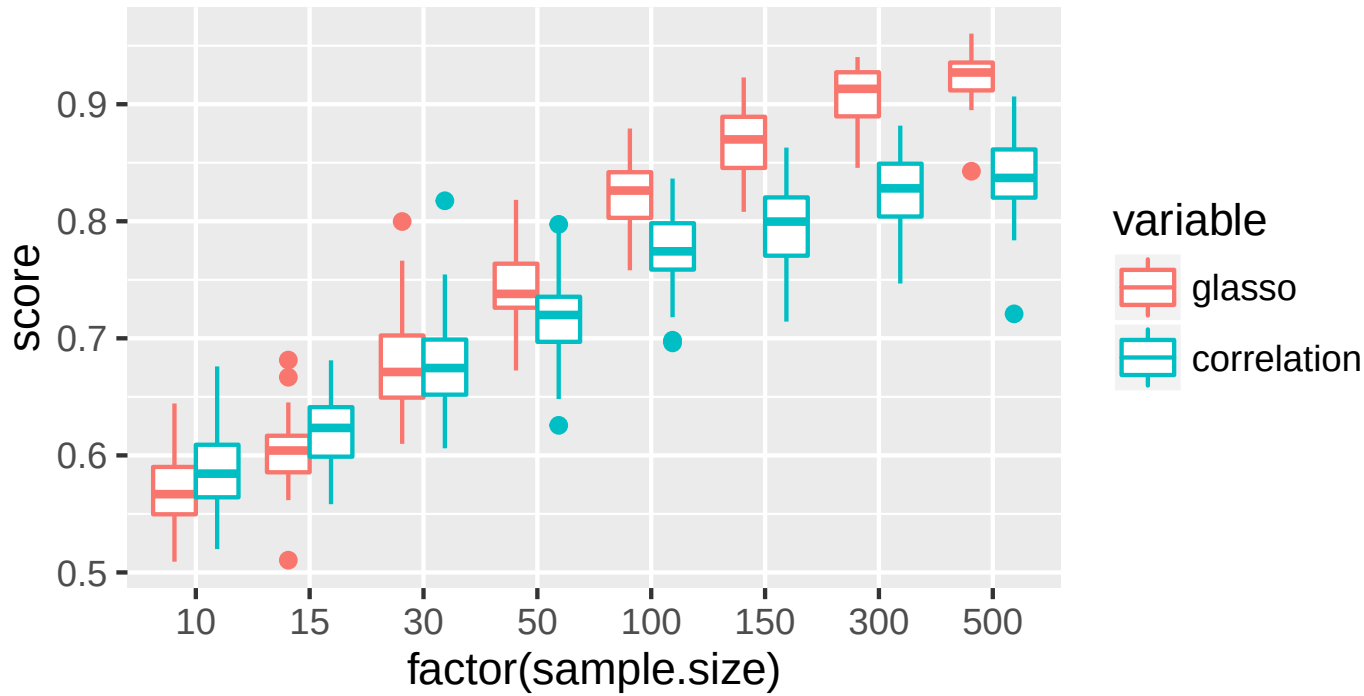
Simulation results (cluster, connection probability of 0.5)

```
suppressMessages(library(ggplot2))
ggplot(res_rd, aes(x=factor(sample.size), y=score)) +
#geom_point(alpha=.5, aes(group=variable,colour=variable))
geom_boxplot(aes(colour=variable))
```



Simulation results (random, connection probability of 0.3)

```
suppressMessages(library(ggplot2))
ggplot(res_rd, aes(x=factor(sample.size), y=score)) +
#geom_point(alpha=.5, aes(group=variable,colour=variable))
geom_boxplot(aes(colour=variable))
```



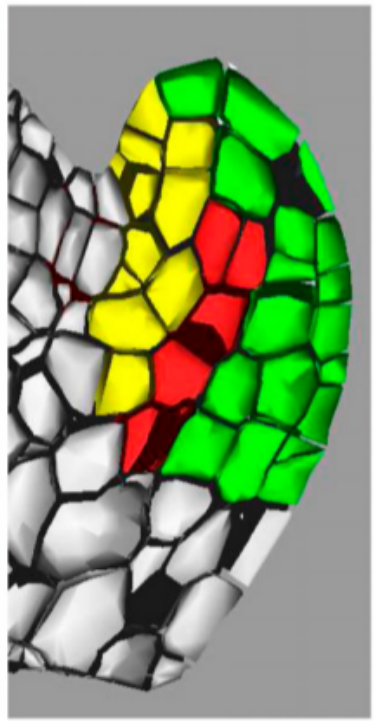
An application to a well defined biological problem

A well characterized network (La Rota et al. 2011)

Construction of a **sepal primordium** network

- Extensive literature/database search
- Expression pattern of different zones of the sepal primordium

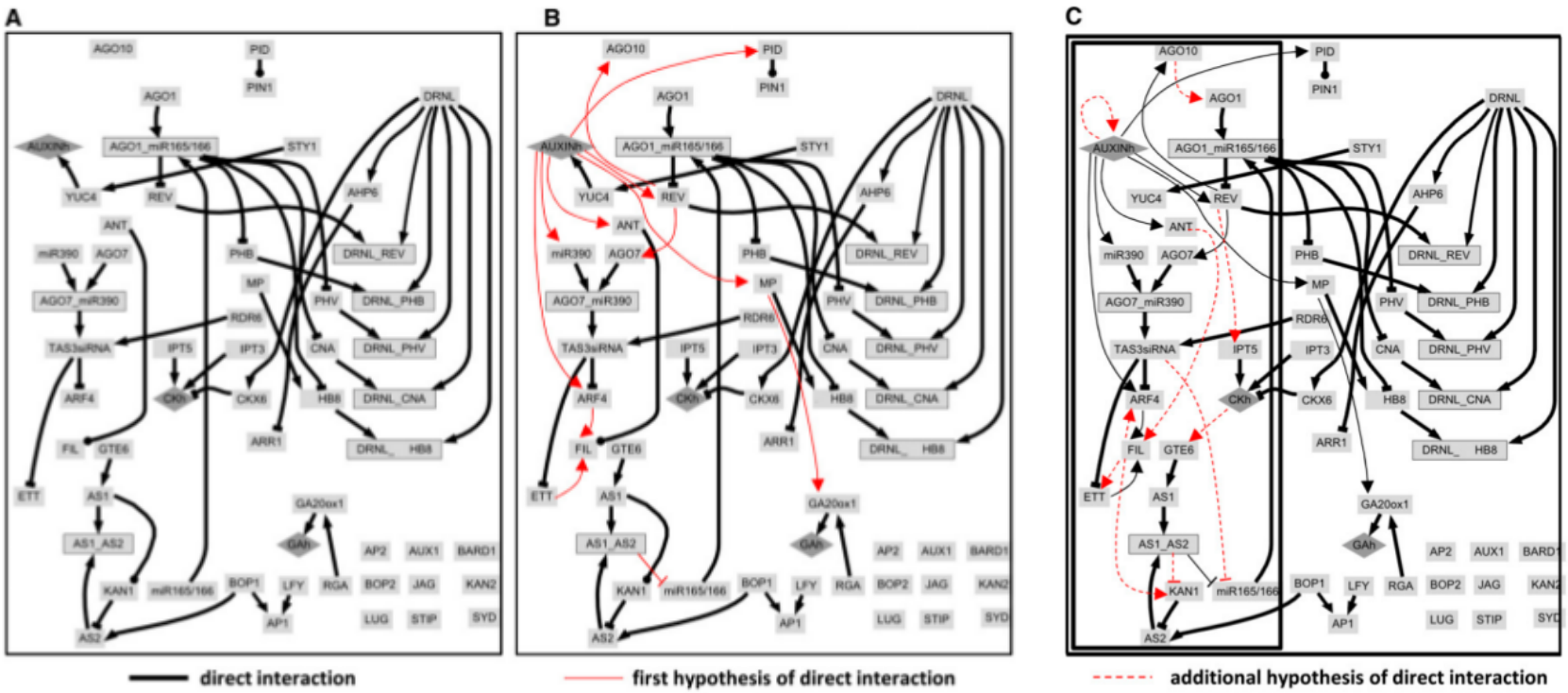
	Zone 1 Adaxial	Zone 2 Vascular	Zone 3 Abaxial
AGO1	1	1	1
AGO10	1	0	0
AGO7	1	0	0
ANT	1	1	1
ARF4	0	1	1
AS1	1	1	0
AS2	1	0	0
ETT	0	1	1
FIL	0	1	1
KAN1	0	0	1
MiR165/166	0	1	1
MiR390	1	1	1
REV	1	0	0
TAS3	1	0	0



A well characterized network

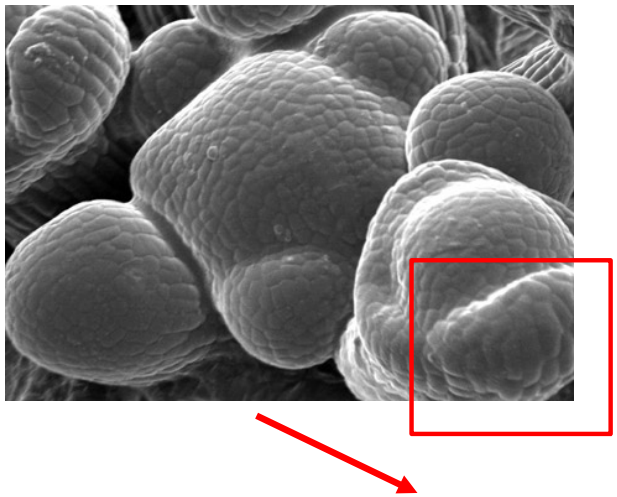
Construction of a **sepal primordium** network

- Successive refinement of the network
- Coherence with observed expression patterns (zone of the sepal primordium)

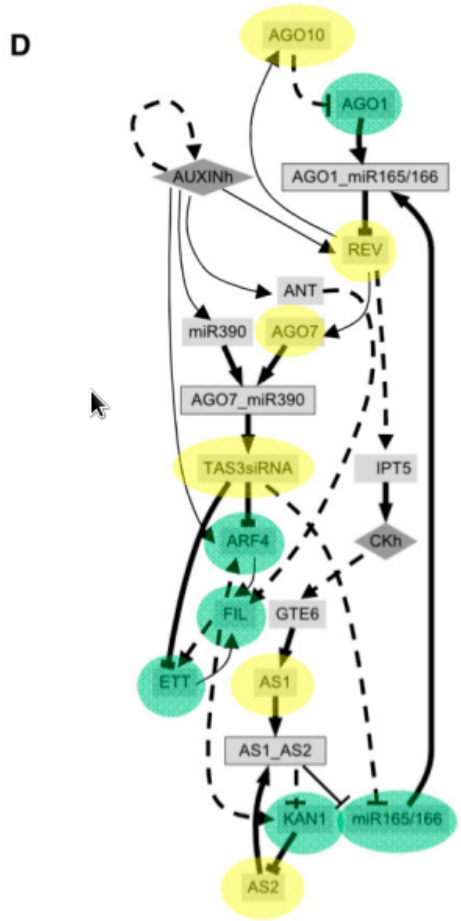


Can we recover part of this network ?

1. Transcriptomic profiles
   - 2×10 samples
   - 16 genes
   - a biologically known network



sépale au stade 3



Analysis with huge

1. Load the data

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

2. Run cor, glasso and glasso with non-paranormal transformation

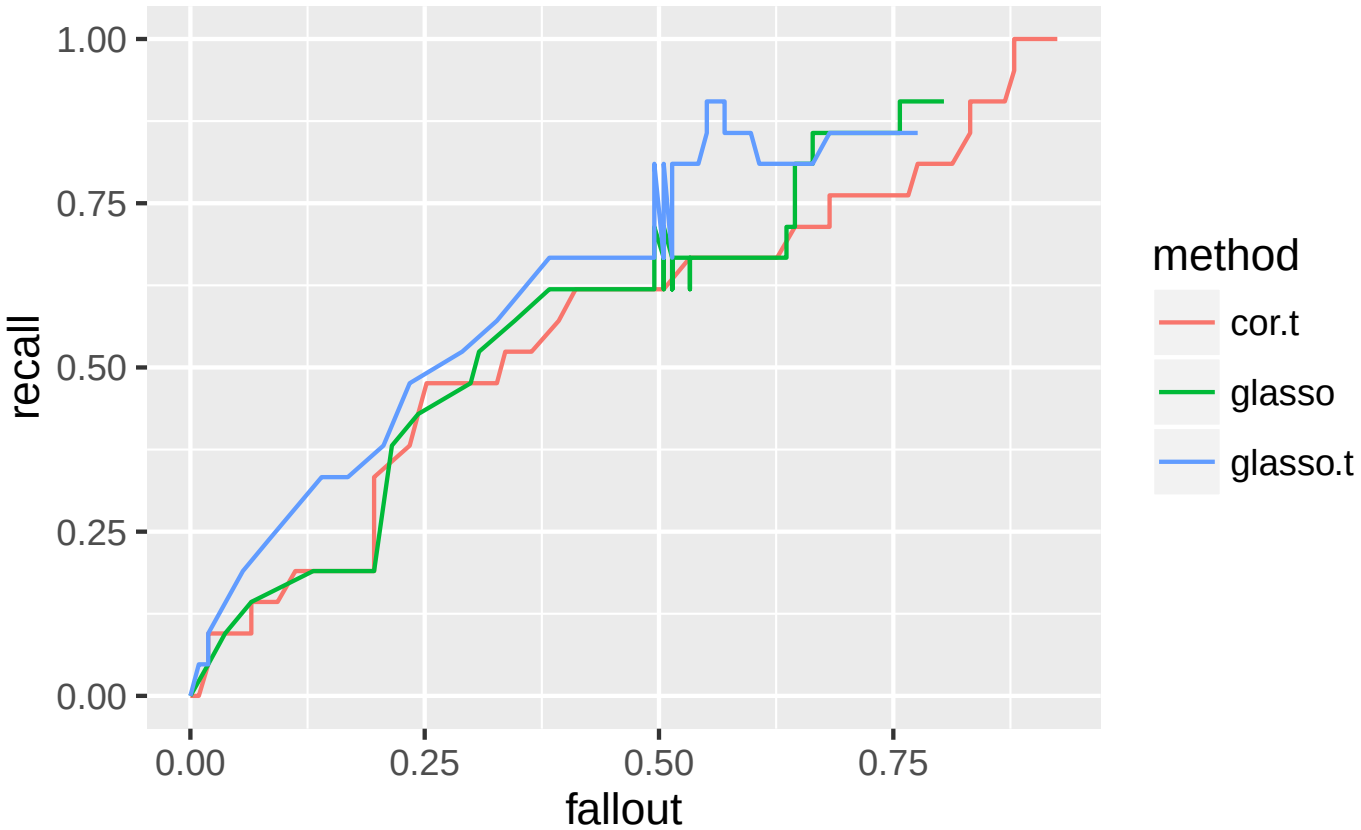
```
Q=huge.npn(X, npn.func="truncation", verbose=F)

lambdas <- rev(1/100 * 10^seq(0, 2,
                              length.out=100))
res.glasso.t <- huge((Q), method="glasso",
                     lambda=lambdas, verbose=F)
res.glasso <- huge((X), method="glasso",
                   lambda=lambdas, verbose=F)

lbd.c <- seq(1, 0, by=-10^-2)
res.cor.t <- huge(X, method="ct",
                  lambda=lbd.c, verbose=F)
```

Transformation is important

```
data <- rbind(roc.glasso.t, roc.glasso, roc.cor.t)
ggplot(data, aes(x=fallout, y=recall, group=method)) +
    geom_line(aes(colour = method))
```



An application to a “less well” defined biological problem

Breast cancer ER+ and ER-

- 1st cancer in women $\sim$ 50000 new cases per year in France
- Very heterogenous disease
- Two main subgroups
  1. Estrogen receptor positive
  2. Estrogen receptor negative

Transcriptome data for ER+ and ER- tumors

We will look at a fairly large public datasets from Guedj et al. 2011:

```
load ("data_school/breast_cancer_guedj11.RData")
load ("data_school/gen.name_.RData")
gene.name <- unlist(gene.name)
data.raw <- expr

table(class.ER)
```

```
## class.ER
## ERm ERp
## 162 375
```

Filtering Not.Known genes

```
toDiscard <- which(gene.name == "Not.Known")
gene.name <- gene.name[-toDiscard]
data.raw <- data.raw[-toDiscard, ]
```

We get

```
dim(data.raw)
```

```
## [1] 41248   537
```

Differential analysis

3. Do we detect some gene expression differences ?

```
suppressMessages(library(limma))
design <- cbind(Moy=1, Erp=(class.ER == "ERp")+0)
fit <- lmFit(data.raw, design=design)
fit <- eBayes(fit)
res <- topTable(fit, coef="Erp", number=10^5,
                  genelist=fit$genes, adjust.method="BH",
                sort.by="none", resort.by=NULL,
                p.value=1, lfc=0, confint=FALSE)
```

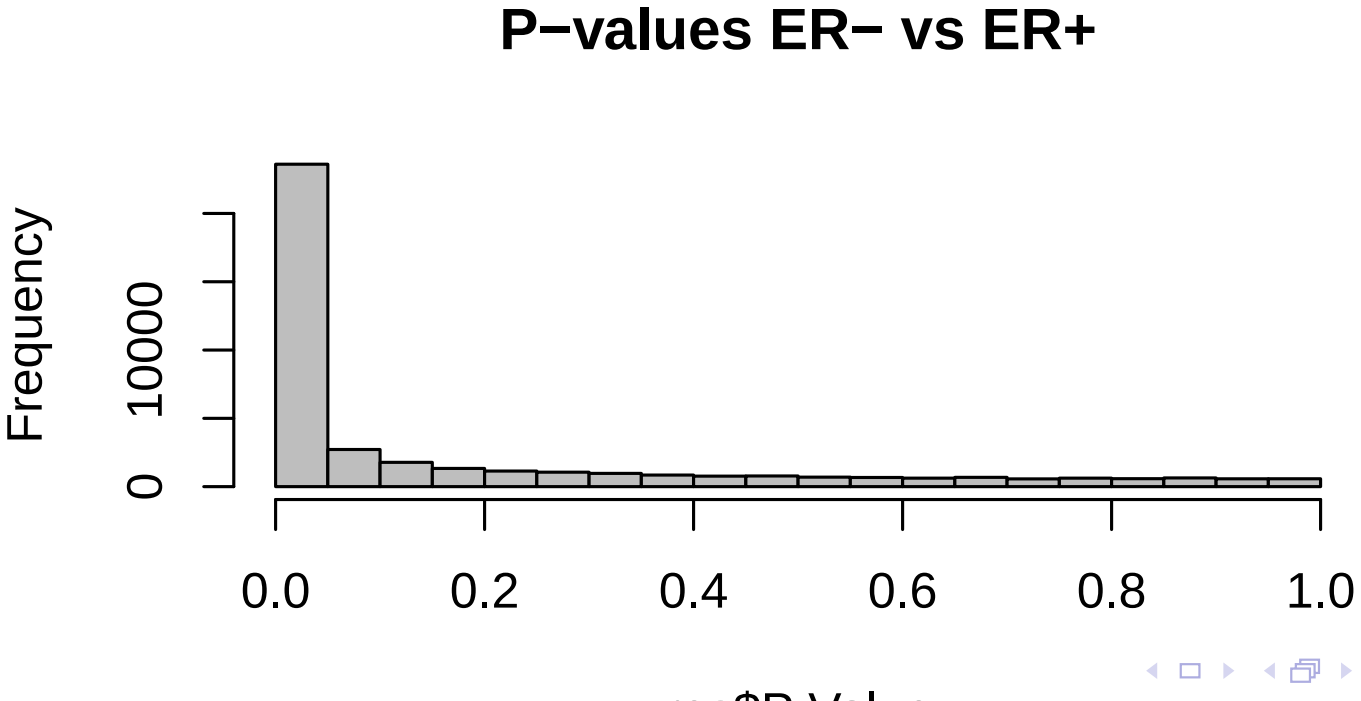
Many genes are differentially expressed

1. The histogram of p-values looks good
2. This is a well known fact (ER+ and ER- are very different)

```
sum(res$adj.P.Val < 10^-5)
```

```
## [1] 5907
```

```
hist(res$P.Value, breaks=30, col="grey",
     main="P-values ER- vs ER+")
```



What to do with this list of genes ?

ESR1 has the most significant p-values

```
gene.name[order(res$adj.P.Val)[1]]
```

```
## 205225_at
##    "ESR1"
```

Network analysis

- Could we find partners of ESR1 that are specific to ER+ ?
- We cannot infer a network on 41000 genes (Verzelen 2011)
  - **Most differentially expressed** genes
  - Most varying genes
  - Look at a specif pathway...

Selecting some probes

1. Take the 20 most differentially expressed, removing duplicated genes (several probes per gene)

```
selected_gene <- order(res$P.Value)[1:200]
sel.name <- gene.name[selected_gene]
selected_gene <- selected_gene[match(unique(sel.name),
                                     sel.name)][1:20]
sel.name <- gene.name[selected_gene]
```

2. We get two small datasets (ER+/ER-)

```
data.ERm <- t(data.raw[selected_gene,
                       which(class.ER=="ERm")])
data.ERp <- t(data.raw[selected_gene,
                       which(class.ER=="ERp")])
```

Network inference

1. Infer the network using glasso.
2. Model selection with stars
3. Check that the optimal index is not at a border

```
lambdas <- rev(1/100 * 10^seq(0, 2, length.out=100))
gl.ERm <- huge(huge.npn(data.ERm, npn.func="truncation", ve
               method="glasso", lambda=lambdas, verbose=F)
gl.ERp <- huge(huge.npn(data.ERp, npn.func="truncation", ve
               method="glasso", lambda=lambdas, verbose=F)
sel.ERm <- huge.select(gl.ERm, criterion="stars",
                       verbose=F)
sel.ERp <- huge.select(gl.ERp, criterion="stars",
                       verbose=F)

sel.ERp$opt.index; sel.ERm$opt.index
```

```
## [1] 18
```

```
## [1] 7
```

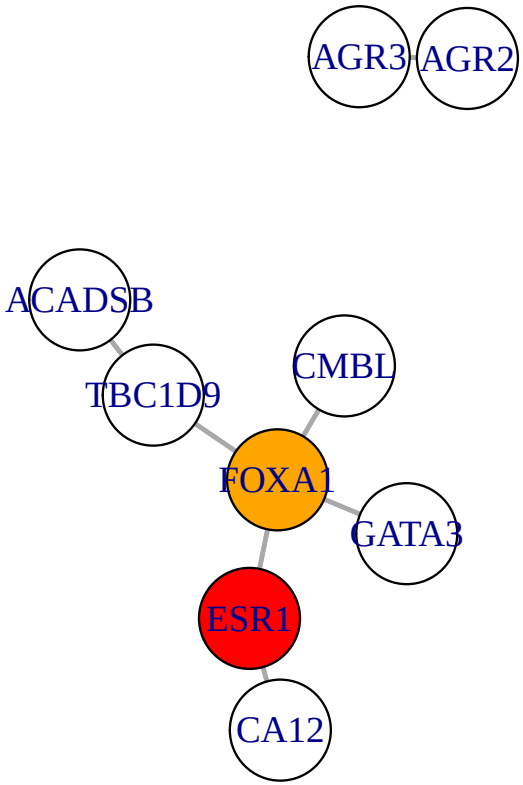
Recover what is specific to each network

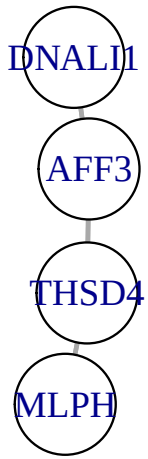
1. Pick edges that are only found in ER+
2. Pick edges that are only found in ER-
3. Remove unecessary node for plotting

```
net_Mspec_ <- as.matrix(sel.ERm$refit) &
  !(as.matrix(sel.ERp$refit))
net_Pspec_ <- as.matrix(sel.ERp$refit) &
  !(as.matrix(sel.ERm$refit))
```

Obtained spectific networks

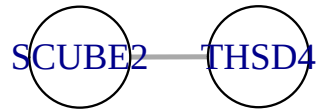
ER+ specific

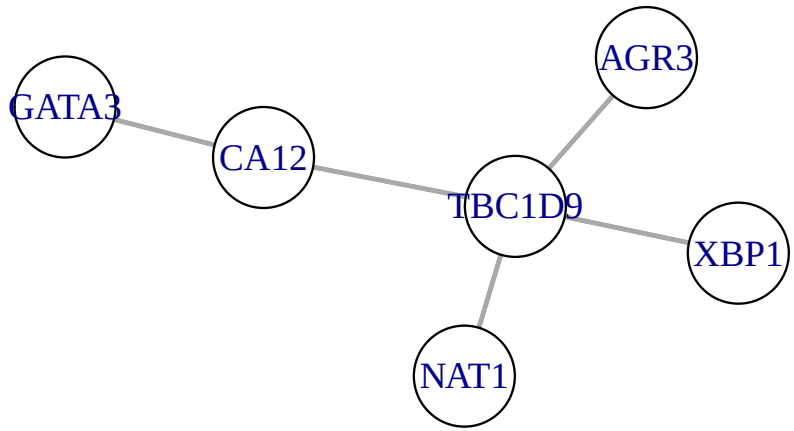




&

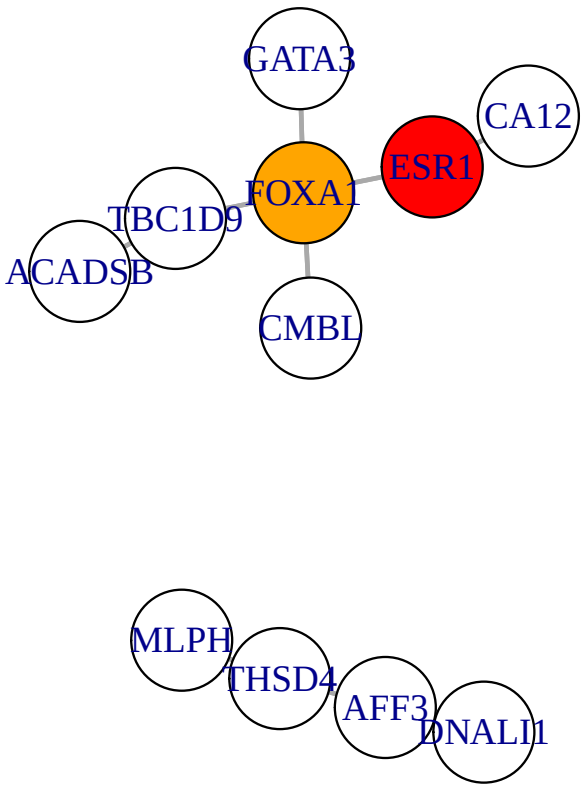
ER− specific

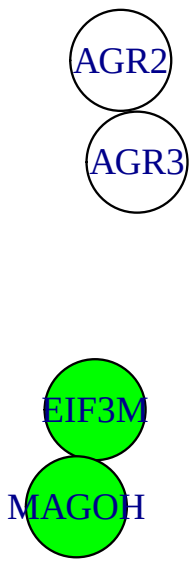




Same thing including some randomly selected genes

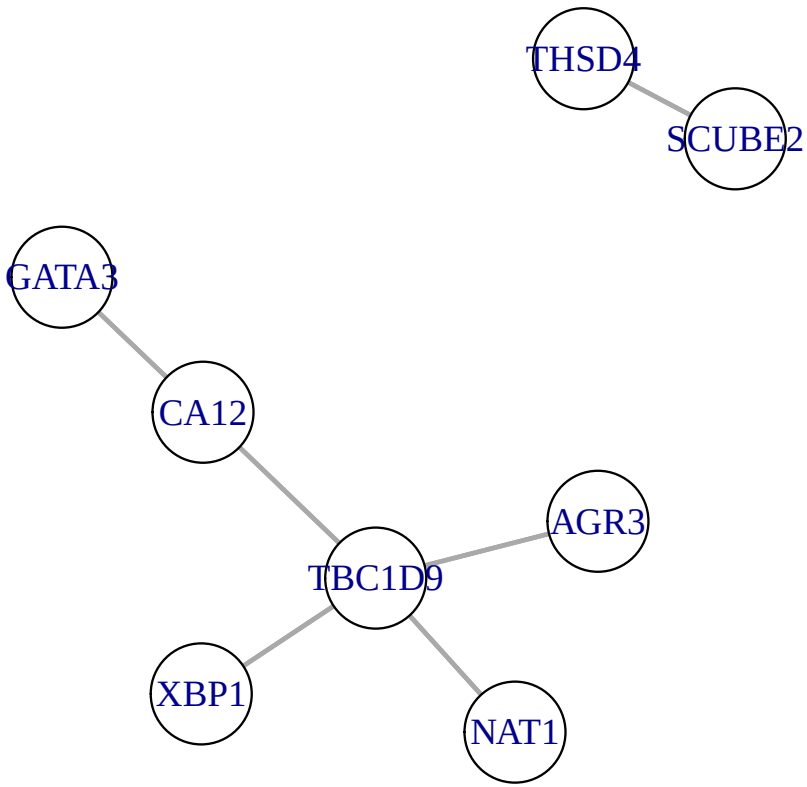
ER+ specific





&

ER− specific



FOXA1, ESR1, GATA3 a well known interaction

1. FOXA1 is a key determinant of estrogen receptor function and endocrine response. Antoni Hurtado et al. 2011 (Nat. Genet.):
   - "FOXA1 is a key determinant that can influence differential interactions between ER and chromatin"
2. GATA3 acts upstream of FOXA1 in mediating ESR1 binding by shaping enhancer accessibility. Theodorou et al. 2013 (Genome Res.)
3. Estrogen receptor regulation of carbonic anhydrase XII through a distal enhancer in breast cancer. Barnett DH et al 2008 (Cancer Res.)
   - "we show that CA12 is robustly regulated by estrogen via ER alpha in breast cancer cells"

Could we find this change with correlation ?

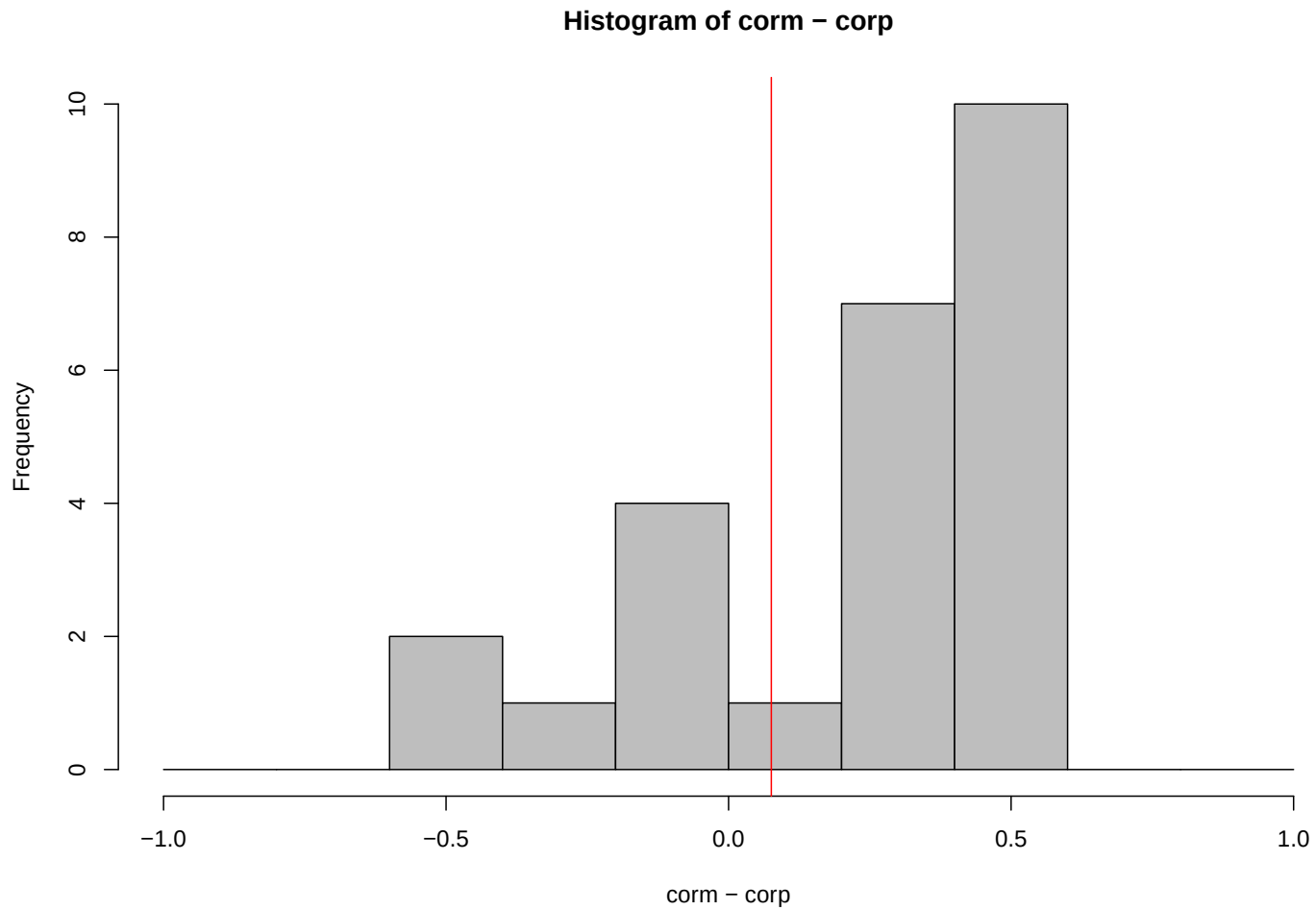
We compute the correlation of ESR1 with other genes

```
corp <- cor(data.ERp, data.ERp[, 1])
corm <- cor(data.ERm, data.ERm[, 1])

iFOXA1 <- which(sel.name == "FOXA1")
col <- rep("black", length(corp))
col[iFOXA1] <- "red"
```

Could we find this change with correlation ?

```
hist(corm-corp, breaks=seq(-1, 1, by=0.2), col="grey")
abline(v=corm[iFOXA1]-corp[iFOXA1], col=c("red"))
```



Some conclusion

1. Some easy to use packages for GGM network inference
2. Choice of the lambda grid not so easy in practise
3. Transformations
4. Efficient for small networks (enough data)
5. Goal
   - Infer the network
   - Pin-point interesting genes for further investigations
6. Dedicated models and extensions are needed (see J. Chiquet presentation's)...
   - Analyzing real datasets is a good way to get relevant modeling ideas

Thanks to

- Julien Chiquet
- Marie-Laure Martin Magniette
- Trung Ha
- Loïc Schwaller
- Françoise Monéger
- Thierry Dubois
- many others
- **and to you for you patience**