Training course 1: Introduction to Gaussian graphical model

Marie Denis

August 10, 2020

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Outline

1. Introduction
   - Definition
   - Gaussian graphical model
2. BDgraph package
3. Application
   - Simulated data
   - Real data
4. To go further
5. Bibliography

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Introduction

Objectives are:

- to introduce the concept of Gaussian graphical model,
- to introduce the R package `BDgraph`,
- to apply on a subset of your data.

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

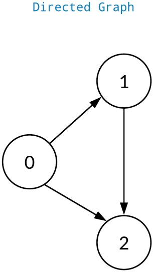
Outline

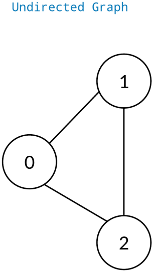
1. Introduction
   - Definition
   - Gaussian graphical model
2. BDgraph package
3. Application
   - Simulated data
   - Real data
4. To go further
5. Bibliography

This project has received funding from
The European Unions's Horizon 2020
Research and Innovation programme
Under grant agreement No 840383.

Some definitions

- A **graph** is a pair $\mathcal{G} = (V, E)$ where $V = \{1, \dots, p\}$ is a finite set of vertices (nodes), and the set of edges E is a subset of the set $V \times V$. Two types of commonly used graphs:
  - **directed graph** where edges are denoted by ordered pairs $(i, j) \in E$
  - **undirected graph** where edges are denoted by unordered pairs $(i, j) \in E \iff (j, i) \in E$.





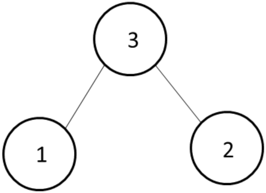
This project has received funding from The European Unions's Horizon 2020 Research and innovation programme Under grant agreement No 840383.

For an overview on graphical models in statistics see Lauritzen (1996).

Some definitions

- Two random variables X_1 and X_2 are **conditionally independent** given the variable X_3 if and only if

$$P(X_1 \cap X_2 | X_3) = P(X_1|X_3)P(X_2|X_3),$$

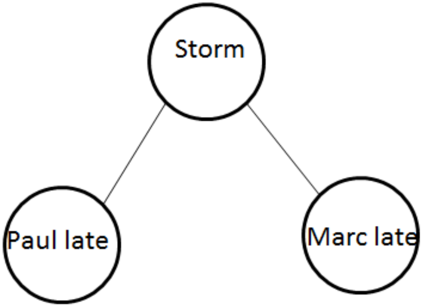


We note $X_1 \perp X_2 | X_3$.

This project has received funding from The European Unions's Horizon 2020 Research and innovation programme Under grant agreement No 840383.

Some definitions

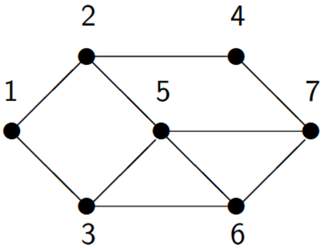
Example:



↬ Given there is a storm renders the event "Paul late" irrelevant for predicting the event "Marc late".

This project has received funding from The European Unions's Horizon 2020 Research and innovation programme Under grant agreement No 840383.

Some definitions



More complex systems of conditional independence can be described by using **Graphical models** (GM).

This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

Graphical model

Graphical models (Lauritzen, 1996) use graph structures for modeling and making statistical inference about **complex relationships among many variables**.

Structure learning refers to the problem of estimating unknown graphs from the data.

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Graphical model

In the following, we will focus on **undirected graphical models** to represent conditional dependence relationships among random variables. In this class of models:

- a vertex is a random variable,
- no edge among two vertices means that these two variables are conditionally independent given the remaining variables (**pairwise Markov property**),
- edge among two vertices means that these two variables are conditionally dependent.

This project has received funding from The European Unions's Horizon 2020 Research and innovation programme Under grant agreement No 840383.

Outline

1. Introduction
   - Definition
   - Gaussian graphical model
2. BDgraph package
3. Application
   - Simulated data
   - Real data
4. To go further
5. Bibliography

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Gaussian graphical model

Gaussian graphical model (GGM) (Dempster, 1972) is a graphical model assuming that the observed data follow a multivariate Gaussian distribution.

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Gaussian graphical model

Multivariate Gaussian distribution:

Let X_i be the p-vector of observed data for subject i such that

$$X_i \sim \mathcal{N}_p(\mu, \Sigma),\ i = 1, \ldots n,$$

with $\mu \in \mathbb{R}^p$ is the mean vector, Σ is the covariance matrix which is a positive semi-definite symmetric matrix, and $\Omega = \Sigma^{-1} \in \mathbb{R}^p \times \mathbb{R}^p$ is the precision matrix.

The **partial correlation** among X^j and X^k given all others is given by $\rho_{X^j X^k, V\setminus\{X^j, X^k\}} = -\frac{\Omega_{jk}}{\sqrt{\Omega_{jj}\Omega_{kk}}}$ where $\Omega = (\Omega_{jk})$.

This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

Gaussian graphical model

Under a Gaussian assumption, conditional independence is implied by the form of the precision matrix. Variables X^j and X^k are conditionally independent given the remaining variables, if and only if $\Omega_{jk} = 0$.

- Undirected graphs are determined by non zeros in the precision matrix Ω (Dempster, 1972; Wang et al., 2015)
- Conditional independence relationships correspond to constraints on the precision matrix Ω, i.e assumption of **sparsity** on Ω.

This project has received funding from The European Unions's Horizon 2020 Research and innovation programme Under grant agreement No 840383.

Estimation techniques

Many of the estimation techniques rely on the assumption of sparsity on the precision matrix:

- Penalized likelihood approaches: graphical Lasso (Friedman et al., 2008), graphical SCAD (Fan et al., 2009), etc.
- Bayesian framework: Bayesian graphical lasso (Wang et al., 2012), graphical horseshoe (Li et al., 2019), etc.

This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

Outline

1. Introduction
   - Definition
   - Gaussian graphical model
2. BDgraph package
3. Application
   - Simulated data
   - Real data
4. To go further
5. Bibliography

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

The BDgraph package

An R package developed by Mohammadi and Wit (2015) for Bayesian structure learning in undirected graphical models.

- **Type of data**: continuous, discrete, and mixed variables,
- **Type of models**: Gaussian graphical model for Gaussian data, and Gaussian copula graphical model (Dobra and Lenkoski 2011) for non-Gaussian, discrete or mixed data,
- **Inference algorithms**: a computationally efficient birth-death Markov chain Monte Carlo (MCMC) sampling algorithm that explores the graph space using birth/death moves to add/remove links.

In the following we will focus on GGM.

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

The model

- Let $G = (V, E)$ an undirected graph, $\mathcal{W} = \{(i,j)|i,j \in V, i < j\}$, and $\bar{E} = \mathcal{W}|E$ the set of non-existing edges,
- Let $\mathcal{M}_G = \{\mathcal{N}_p(0, \Sigma)|\Omega = \Sigma^{-1} \in \mathbb{P}_G\}$ be a zero mean GGM, where $\mathbb{P}_G$ denotes the space of $p \times p$ positive definite matrices with entries (i,j) equal to zero whenever $(i,j) \in \bar{E}$,
- Let $X = (X_1, \dots, X_n)$ be an independent and identically distributed sample of size n from model $\mathcal{M}_G$ with the following likelihood:

$$p(X|\Omega, G) \propto |\Omega|^{n/2} exp(-\frac{1}{2} tr(\Omega X' X))$$

Parameters to estimate: G and Ω.

Function to use to perform models: `bdgraph`.

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Bayesian framework

Objective is to sample from the joint posterior distribution $p(G, \Omega | X)$ known to a normalizing constant.

From the Bayes theorem:

$$p(G, \Omega | X) \propto p(X | G, \Omega) p(\Omega | G) p(G)$$

We need to specify:

- Prior distribution for $\Omega | G$: G-Wishart distribution
- Prior distribution for G: Bernoulli prior on each link inclusion indicator variable as follow
$$Pr(G) \propto (\theta)^{|E|} (1-\theta)^{\binom{n}{2} - |E|}$$
with $|E|$ the number of links in the graph G (graph size), and $\theta \in (0,1)$ the prior probability of a link.

MCMC sampling algorithm is used to sample from $p(G, \Omega | X)$.

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Convergence check

Need to check the convergence of MCMC chains to the target joint posterior distribution

- Theoretically the sampling distribution will converge to the target joint posterior distribution as the number of iterations increases to infinity
- Two typical questions arise: how many MCMC samples are sufficient? and how long should the ”burn-in” period be?

Functions to use: `plotcoda` and `traceplot`

This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

Posterior graph selection

Selection may be based on:

- Bayesian model averaging (BMA) (default approach) ↬ inferred graph is a graph with links for which the estimated posterior probabilities are greater than a certain cut-point,
- Maximum a posterior probability (MAP) ↬ inferred graph is a graph with the highest posterior probability.

Functions to use: `select` and `plinks`

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Outline

1. Introduction
   - Definition
   - Gaussian graphical model

2. BDgraph package

3. Application
   - Simulated data
   - Real data

4. To go further

5. Bibliography

This project has received funding from The European Unions's Horizon 2020 Research and innovation programme Under grant agreement No 840383.

Outline

1. Introduction
   - Definition
   - Gaussian graphical model
2. BDgraph package
3. Application
   - Simulated data
   - Real data
4. To go further
5. Bibliography

This project has received funding from The European Unions's Horizon 2020 Research and innovation programme Under grant agreement No 840383.

The main function

The main function bdgraph

```
> library(BDgraph)#To download the package
> bdgraph(data,#an (nxp) matrix or a data.frame or a covariance (pxp) matri
+     n = NULL,#the sample size
+     method = "ggm",#method: "ggm" for GGM or "gcgm" GCGM
+     iter = 5000,burnin = iter/2,#nb of iterations, and burn-in
+     g.prior = 0.5,#parameter of Bernoulli prior
+     g.start = "empty",#the initial graph for sampling algorithm
+     jump = NULL,#number of links that are simultaneously updated
+     #in the BDMCMC algorithm
+     save = FALSE,# to save the samples
+     print = 1000,
+     cores = NULL,#the number of cores to use for parallel execution
+     )
```

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Simulated data

We start by simulating Gaussian multivariate data with the function `bdgraph.sim()`. The sample size is equal to $n = 200$, and the number of variables to $p = 8$.

Simulation

```
> set.seed(1)
> data.sim <- bdgraph.sim( n = 200, p = 8,
+                          graph = "random",
+                          prob = 0.4,
+                          type = "Gaussian" )
> str(data.sim)
> #round(head(data.sim$data,4), 2 )
```

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Analysis

We use the function bdgraph to perfom the model.

Analysis

```
> set.seed(1)
> sample.bdmcmc <-  bdgraph( data = data.sim$data,
+                            method = "ggm",
+                            algorithm = "bdmcmc",
+                            g.prior = 0.4,
+                            iter = 5000, burnin = 0,
+                            save = TRUE, print = 1000)
```

This project has received funding from
The European Unions's Horizon 2020
Research and Innovation programme
Under grant agreement No 840383.

Results

We extract results with the function summary.

Summary

```
> summary(sample.bdmcmc)
> str(summary(sample.bdmcmc))
```

Three outputs are obtained:

- selected_g: the adjacency matrix of the selected graph based on BMA estimations,
- p_links: the estimated posterior probabilities of all possible links,
- K_hat: the estimated precision matrix.

We can select graph with the function select:

```
> BDgraph::select(sample.bdmcmc,cut = 0.4, vis = FALSE)
```

This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

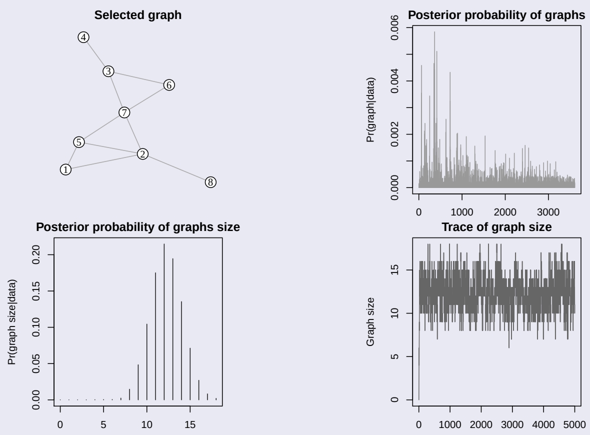
Plots I

Four plots are also generated.

- On top-left: selected graph from BMA estimations,
- On top-right: estimated posterior probabilities of all visited graphs,
- On bottom-left: estimated probabilities of the size of graphs,
- On bottom-right: trace of our algorithm based on the size of the graphs.

This project has received funding from
The European Unions's Horizon 2020
Research and Innovation programme
Under grant agreement No 840383.

Plots II



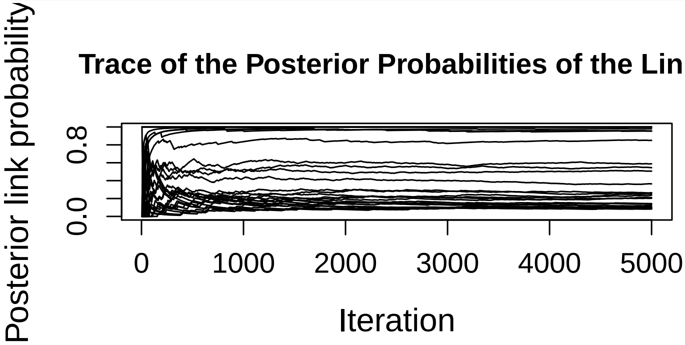
This project has received funding from The European Unions's Horizon 2020 Research and innovation programme Under grant agreement No 840383.

Convergence check

We used the function `plotcoda` to check if the number of iterations is enough, and to monitor the convergence of the MCMC algorithm.

Convergence check

```
> plotcoda(sample.bdmcmc)
```



This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

To compare with the truth

We use the function compare.

To compare with the truth

```
> BDgraph::compare(data.sim, sample.bdmcmc,
+                  main = c( "Target", "BDgraph" ))

               Target BDgraph
true positive      12   9.000
true negative      16  15.000
false positive      0   1.000
false negative      0   3.000
F1-score            1   0.818
specificity         1   0.938
sensitivity         1   0.750
MCC                 1   0.710
```

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

To compare with the graphical Lasso (Friedman et al. 2007)

We use the function `glassopath` from the package `glasso`.

To compare with the truth

```
> s <- var(data.sim$data)
> a <- glassopath(s)
> i <- 1
> a$wi[,,i][a$wi[,,i] !=0] <- 1
> BDgraph::compare( data.sim,a$wi[,,i],
+                   main = c( "Target", "Graphical Lasso" ))

               Target Graphical Lasso
true positive      12           6.000
true negative      16          16.000
false positive      0           0.000
false negative      0           6.000
F1-score            1           0.667
specificity         1           1.000
sensitivity         1           0.500
MCC                 1           0.603
```

Outline

1. Introduction
   - Definition
   - Gaussian graphical model
2. BDgraph package
3. Application
   - Simulated data
   - Real data
4. To go further
5. Bibliography

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Data description

- Using the MOTA data sent by Ziling, a subset of mRNAs from the GU2 cohort were selected by applying a sparse PCA.

↬ A total of 50 genes for 61 patients (37 cases and 24 controls) were analyzed.

- Somes gene expressions were equal to 0 ⇒ we put to 0.01,
- A log2 transformation was applied.

To download data

```
> gene <- read.csv(file = "gene_lesson1.csv",
+                  header = T, sep = ";")
> status <- as.matrix(read.csv(file = "Y_lesson1.csv",
+                              header = T, sep = ";"))
```

This project has received funding from
The European Unions's Horizon 2020
Research and Innovation programme
Under grant agreement No 840383.

PCA analysis

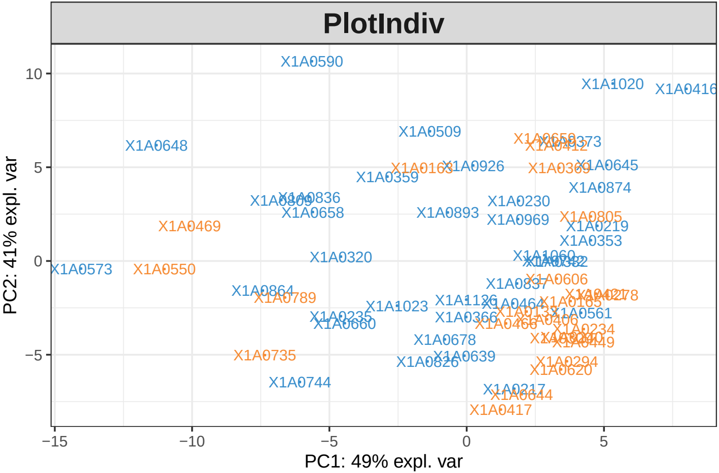
I used PCA to perform an explanatory data analysis on the subsetted data using the function `pca` from the `mixOmics` package. Plots of individuals and variables are obtained with the functions `plotIndiv` and `plotVar`, respectively.

PCA analysis

```
> library(mixOmics)
> pca.gene <- pca(gene, ncomp=4, center = TRUE, scale = TRUE)
> plotIndiv(pca.gene, comp = c(1,2), group = status)
> plotVar(pca.gene, comp = c(1,2), var.names = TRUE)
```

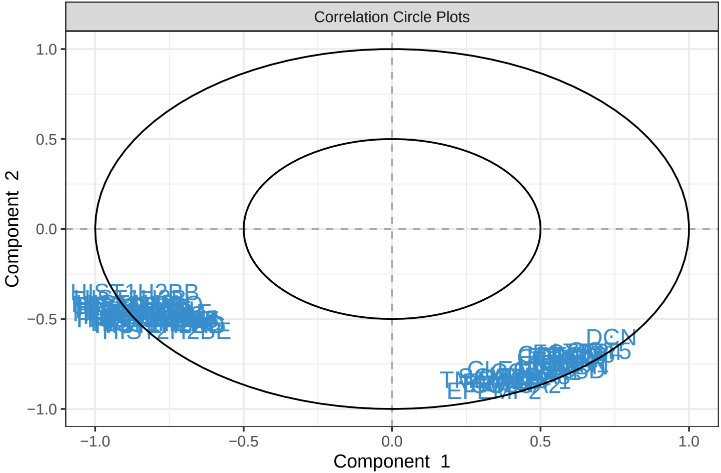
This project has received funding from
The European Unions's Horizon 2020
Research and Innovation programme
Under grant agreement No 840383.

Individual plot



This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

Variable plot



This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

Gaussian graphical model

We use the function `bdgraph` to perform a Gaussian gaphical model.

Analysis

```
> set.seed(1)
> bdgraph.obj <-  bdgraph(data = scale(gene),
+                         n = nrow(gene),
+                         g.prior = 0.1,
+                         iter = 10000,
+                         burnin = 1,
+                         save = TRUE, print = 1000)
```

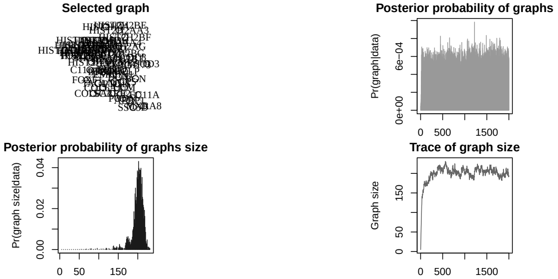
This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Gaussian graphical model I

We extract results with the function summary.

Results

```
> res.bdgraph.obj <- summary(bdgraph.obj)
> BDgraph::select(bdgraph.obj,cut = 0.5, vis = FALSE)
```



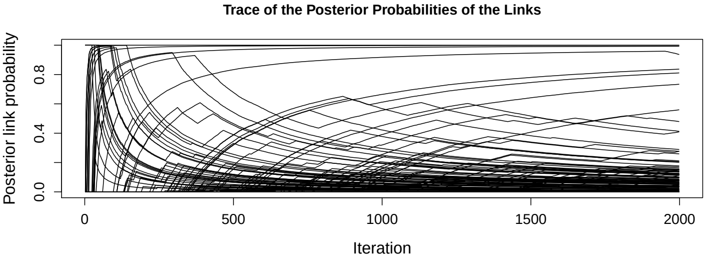
This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

Gaussian graphical model

We monitor the convergence with the function `plotcoda`.

Results

```
> plotcoda(bdgraph.obj)
```



This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

From these results

We use the function cov2cor to obtain the partial correlations and the function graph.adjacency (from the igraph package) to obtain an object of class igraph.

Partial correlation

```
> prec <- res.bdgraph.obj$K_hat # precision matrix
> cor.par <- -cov2cor(prec) # partial correlations
> cor.par[res.bdgraph.obj$selected_g ==0] <- 0
> gadj <- graph.adjacency(cor.par,weighted = TRUE,
+                         mode ="undirected",
+                         diag = FALSE)
```

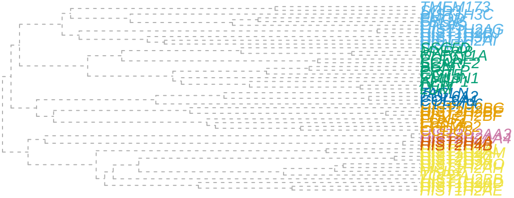
This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

From these results

We use the functions `cluster_fast_greedy` and `dendPlot` from the package `igraph` to cluster variables and to produce a dendogram.

Clustering and dendogram

```
> E(gadj)$weight <- abs(E(gadj)$weight)
> clp <- cluster_fast_greedy(gadj)
> dendPlot(clp) #, mode="hclust"
> plot(clp, gadj)
```



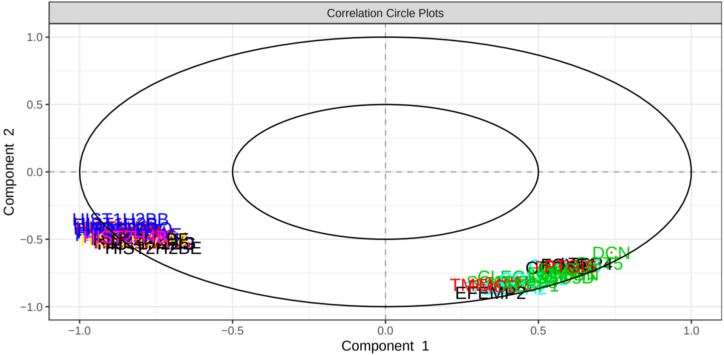
This project has received funding from The European Unions's Horizon 2020 Research and innovation programme Under grant agreement No 840383.

From these results

We use memberships to clusters obtained previously to color genes in PCA plot.

PCA again

```
> plotVar(pca.gene, comp = c(1,2), var.names = TRUE,
+          col = list(clp$membership))
```



This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

Outline

1. Introduction
   - Definition
   - Gaussian graphical model
2. BDgraph package
3. Application
   - Simulated data
   - Real data
4. To go further
5. Bibliography

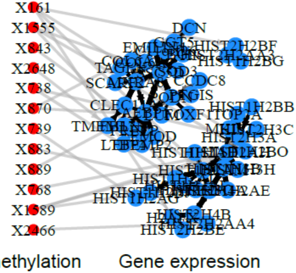
This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

More than one dataset

Multivariate Bayesian variable and covariance selection models, as proposed in the `BayesSUR` package (Banterlee et al., 2020), allow to analyze simultaneously two datasets and to select relevant markers.

Methylation and mRNA datasets

Based on the same subset of genes analyzed previously, and using a subset of methylations selected with a sparse PCA, the results show links among responses, and among responses and predictors:

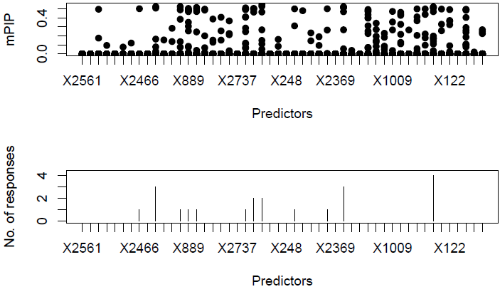


This project has received funding from The European Unions's Horizon 2020 Research and innovation programme Under grant agreement No 840383.

More than one dataset

Methylation and mRNA datasets

Plots represent the marginal inclusion probability for each predictor (top) and the number of responses for which a predictor has been selected (bottom).



This project has received funding from The European Unions's Horizon 2020 Research and Innovation programme Under grant agreement No 840383.

Outline

1 Introduction
- Definition
- Gaussian graphical model

2 BDgraph package

3 Application
- Simulated data
- Real data

4 To go further

5 Bibliography

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.

Dempster, A. P. (1972). Covariance selection. *Biometrics*, pages 157–175.

Fan, J., Feng, Y., and Wu, Y. (2009). Network exploration via the adaptive lasso and scad penalties. *The annals of applied statistics*, 3(2):521.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Lauritzen, S. L. (1996). *Graphical models*, volume 17. Clarendon Press.

Li, Y., Datta, J., Craig, B. A., and Bhadra, A. (2019). Joint mean-covariance estimation via the horseshoe with an application in genomic data analysis. *arXiv preprint arXiv:1903.06768*.

Mohammadi, R. and Wit, E. C. (2015). Bdgraph: An r package for bayesian structure learning in graphical models. *arXiv preprint arXiv:1501.05108*.

Wang, H. et al. (2012). Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis*, 7(4):867–886.

Wang, H. et al. (2015). Scaling it up: Stochastic search structure learning in graphical models. *Bayesian Analysis*, 10(2):351–377.

This project has received funding from
The European Unions's Horizon 2020
Research and innovation programme
Under grant agreement No 840383.