

# Mini-projet 2021 - Module 6 Bioinformatique Intégrative

Magali Monnoye

2021-05-31

## Contents

<b>Synopsis du projet</b>	<b>2</b>
Données . . . . .	2
<b>Import données</b>	<b>3</b>
Download the files . . . . .	3
Loading the files . . . . .	5
<b>Analyses différentielles DESeq2</b>	<b>12</b>
Modèle fa en comparant Day 7 à normal . . . . .	12
Modèle pfa en comparant Day 7 à normal . . . . .	16
<b>MixOmics</b>	<b>20</b>
Préparation des tables . . . . .	20
Filtres . . . . .	22
Import package . . . . .	24
Analyses uni omic . . . . .	24
PCA . . . . .	24
Sparse PCA . . . . .	27
PLS-DA . . . . .	31
Sparse PLS-DA . . . . .	37
Analyses multi omics . . . . .	45
PLS . . . . .	45
S-PLS . . . . .	47
Multi-block PLS-DA . . . . .	51
Multi-block S-PLS-DA / DIABLO . . . . .	53

<b>WGCNA</b>	<b>59</b>
Filtres . . . . .	62
Arbre permettant de détecter les valeurs abérantes . . . . .	63
Import trait table . . . . .	65
Adjacency matrix . . . . .	67
Cluster Dendrogram . . . . .	70
Heatmap relation module/trait . . . . .	72
Eigengene view . . . . .	73
Export Cytoscape . . . . .	75
Remise du rapport . . . . .	76
Session info . . . . .	76
J'importe les librairies	

Table 1: Loaded required libraries

libraries
knitr
FactoMineR
factoextra
gprofiler2
pheatmap
dplyr

## Synopsis du projet

### Données

Pavkovic, M., Pantano, L., Gerlach, C.V. et al. Multi omics analysis of fibrotic kidneys in two mouse models. Sci Data 6, 92 (2019)

Samples from two mouse models were collected. The first one is a reversible chemical-induced injury model (folic acid (FA) induced nephropathy). The second one is an irreversible surgically-induced fibrosis model (unilateral ureteral obstruction (UUO)). mRNA and small RNA sequencing, as well as 10-plex tandem mass tag (TMT) proteomics were performed with kidney samples from different time points over the course of fibrosis development.

### Rappel sur les échantillons:

Deux modèles de fibrose rénale chez la souris sont étudiés:

1. Le premier est un modèle de néphropathie réversible induite par l'acide folique (folic acid (FA)). Les souris ont été sacrifiées avant le traitement (normal), puis à jour 1, 2, 7 et 14 (day1,...) après une seule injection d'acide folique.
2. Le second est un modèle irréversible induit chirurgicalement (unilateral ureteral obstruction (UUO)). les souris ont été sacrifiées avant obstruction (day 0) et à 3, 7 et 14 jours après obstruction par ligation de l'uretère du rein gauche.

A partir de ces extraits de rein, l'ARN messager total et les petits ARNs ont été séquencés et les protéines caractérisées par spectrométrie de masse en tandem (TMT).

Supplementary material of the article with all the data tables (zip archive):

Les données se trouvent aussi dans le dépôt github

Les comptages du modèle FA sont dans tables/fa/results/counts/ pour la transcriptomique tables/pfa/results/counts/ pour la protéomique

Les comptages du modèle UUO sont dans tables/uuo/results/counts/ pour la transcriptomique tables/puuo/results/counts/ pour la protéomique

Nous travaillerons sur le modèle **FA**.

## Import données

Je télécharge les quatre fichiers dans un dossier local ~/Module6Projet, et les charge dans les data.frames suivants:

- Données brutes de transcriptome: `fa_expr`
- Métadonnées transcriptome: `fa_meta`
- Données brutes de protéomique: `pfa_expr`
- Métadonnées transcriptome: `pfa_meta`

J'importe directement les données normalisées

## Download the files

```
'#' @title Download a file only if it is not yet here
#' @author Jacques van Helden email{Jacques.van-Helden@france-bioinformatique.fr}
#' @param url_base base of the URL, that will be prepended to the file name
#' @param file_name name of the file (should not contain any path)
#' @param local_folder path of a local folder where the file should be stored
#' @return the function returns the path of the local file, built from local_folder and file_name
#' @export
downloadOnlyOnce <- function(url_base,
                               file_name,
                               local_folder) {

  ## Define the source URL
  url <- file.path(url_base, file_name)
  message("Source URL\n\t", url)

  ## Define the local file
  local_file <- file.path(local_folder, file_name)

  ## Create the local data folder if it does not exist
  dir.create(local_folder, showWarnings = FALSE, recursive = TRUE)

  ## Download the file ONLY if it is not already there
  if (!file.exists(local_file)) {
```

```

    message("Downloading file from source URL to local file\n\t",
           local_file)
    download.file(url = url, destfile = local_file)
} else {
  message("Local file already exists, no need to download\n\t",
         local_file)
}

return(local_file)
}

```

```

## Specify the basic parameters
pavkovic_base <- "https://github.com/DU-Bii/module-3-Stat-R/tree/master/stat-R_2021/data/pavkovic_2019"
pavkovic_folder <- "~/DUBii-m3_data/pavkovic_2019"

##### Download folic acid data and metadata #####
## Transcriptome data table
local_fa_file <- downloadOnlyOnce(
  url_base = pavkovic_base,
  file_name = "fa_raw_counts.tsv.gz",
  local_folder = pavkovic_folder
)

## Transcriptome data table normalized
local_fa_file_norm <- downloadOnlyOnce(
  url_base = pavkovic_base,
  file_name = "fa_normalized_counts.tsv.gz",
  local_folder = pavkovic_folder
)

## Transcriptome metadata
trans_metadata_file <- downloadOnlyOnce(
  url_base = pavkovic_base,
  file_name = "fa_transcriptome_metadata.tsv",
  local_folder = pavkovic_folder
)

## Proteome data table
local_pfa_file <- downloadOnlyOnce(
  url_base = pavkovic_base,
  file_name = "pfa_model_counts.tsv.gz",
  local_folder = pavkovic_folder
)

## Proteome data table normalized
local_pfa_file_norm <- downloadOnlyOnce(
  url_base = pavkovic_base,
  file_name = "pfa_model_log2_counts.tsv.gz",
  local_folder = pavkovic_folder
)

## Proteome metadata

```

```

prot_metadata_file <- downloadOnlyOnce(
  url_base = pavkovic_base,
  file_name = "pfa_proteome_metadata.tsv",
  local_folder = pavkovic_folder
)

```

## Loading the files

```

#' @title Load a tab-separated value file and manually set row names after having forced them to unique
#' @author Jacques van Helden email{Jacques.van-Helden@france-bioinformatique.fr}
#' @param file file path
#' @param header=1 Header is set to 1 by default
#' @param sep="\t" Column separator is set to tab by default
#' @param rownames.col=1 Column containing the row names
#' @param ... all other parameters are passed to read.delim()
#' @return a data frame with the loaded data
load_fix_row_names <- function(file,
  header = 1,
  sep = "\t",
  rownames.col = 1,
  ...) {
  x <- read.delim(file = file, ...)
  rownames(x) <- make.names(x[, rownames.col], unique = TRUE)
  x <- x[, -rownames.col]
  return(x)
}

```

```

## Load transcriptome data
fa <- read.delim(file = local_fa_file, sep = "\t", header = TRUE)

## Load same data with load_fix_row_names
fa_expr <- load_fix_row_names(file = local_fa_file, rownames.col = 1)
kable(head(fa_expr), caption = "Loaded with myEasyLad() fa")

```

Table 2: Loaded with myEasyLad() fa

day1_d1ay1_d2ay1_d3ay1d4ay1d4d2ay1d3ay2_d1ay2_d2ay2_d3ay3_d1ay3_d2ay3_d3ay7_d1ay7_d2ay7_d3ay7_d4ay7_d5ay7_d6ay7_d7ay7_d8ay7_d9ay7_d10ay7_d11ay7_d12ay7_d13ay7_d14ay7_d15ay7_d16ay7_d17ay7_d18ay7_d19ay7_d20ay7_d21ay7_d22ay7_d23ay7_d24ay7_d25ay7_d26ay7_d27ay7_d28ay7_d29ay7_d30ay7_d31ay7_d32ay7_d33ay7_d34ay7_d35ay7_d36ay7_d37ay7_d38ay7_d39ay7_d40ay7_d41ay7_d42ay7_d43ay7_d44ay7_d45ay7_d46ay7_d47ay7_d48ay7_d49ay7_d50ay7_d51ay7_d52ay7_d53ay7_d54ay7_d55ay7_d56ay7_d57ay7_d58ay7_d59ay7_d60ay7_d61ay7_d62ay7_d63ay7_d64ay7_d65ay7_d66ay7_d67ay7_d68ay7_d69ay7_d70ay7_d71ay7_d72ay7_d73ay7_d74ay7_d75ay7_d76ay7_d77ay7_d78ay7_d79ay7_d80ay7_d81ay7_d82ay7_d83ay7_d84ay7_d85ay7_d86ay7_d87ay7_d88ay7_d89ay7_d90ay7_d91ay7_d92ay7_d93ay7_d94ay7_d95ay7_d96ay7_d97ay7_d98ay7_d99ay7_d100ay7_d101ay7_d102ay7_d103ay7_d104ay7_d105ay7_d106ay7_d107ay7_d108ay7_d109ay7_d110ay7_d111ay7_d112ay7_d113ay7_d114ay7_d115ay7_d116ay7_d117ay7_d118ay7_d119ay7_d120ay7_d121ay7_d122ay7_d123ay7_d124ay7_d125ay7_d126ay7_d127ay7_d128ay7_d129ay7_d130ay7_d131ay7_d132ay7_d133ay7_d134ay7_d135ay7_d136ay7_d137ay7_d138ay7_d139ay7_d140ay7_d141ay7_d142ay7_d143ay7_d144ay7_d145ay7_d146ay7_d147ay7_d148ay7_d149ay7_d150ay7_d151ay7_d152ay7_d153ay7_d154ay7_d155ay7_d156ay7_d157ay7_d158ay7_d159ay7_d160ay7_d161ay7_d162ay7_d163ay7_d164ay7_d165ay7_d166ay7_d167ay7_d168ay7_d169ay7_d170ay7_d171ay7_d172ay7_d173ay7_d174ay7_d175ay7_d176ay7_d177ay7_d178ay7_d179ay7_d180ay7_d181ay7_d182ay7_d183ay7_d184ay7_d185ay7_d186ay7_d187ay7_d188ay7_d189ay7_d189ay7_d190ay7_d191ay7_d192ay7_d193ay7_d194ay7_d195ay7_d196ay7_d197ay7_d198ay7_d199ay7_d199ay7_d200ay7_d201ay7_d202ay7_d203ay7_d204ay7_d205ay7_d206ay7_d207ay7_d208ay7_d209ay7_d209ay7_d210ay7_d211ay7_d212ay7_d213ay7_d214ay7_d215ay7_d216ay7_d217ay7_d218ay7_d219ay7_d219ay7_d220ay7_d221ay7_d222ay7_d223ay7_d224ay7_d225ay7_d226ay7_d227ay7_d228ay7_d229ay7_d229ay7_d230ay7_d231ay7_d232ay7_d233ay7_d234ay7_d235ay7_d236ay7_d237ay7_d237ay7_d238ay7_d239ay7_d239ay7_d240ay7_d241ay7_d242ay7_d243ay7_d244ay7_d245ay7_d246ay7_d247ay7_d247ay7_d248ay7_d249ay7_d249ay7_d250ay7_d251ay7_d252ay7_d253ay7_d254ay7_d255ay7_d256ay7_d257ay7_d257ay7_d258ay7_d259ay7_d259ay7_d260ay7_d261ay7_d262ay7_d263ay7_d264ay7_d265ay7_d266ay7_d267ay7_d267ay7_d268ay7_d269ay7_d269ay7_d270ay7_d271ay7_d272ay7_d273ay7_d274ay7_d275ay7_d276ay7_d277ay7_d277ay7_d278ay7_d279ay7_d279ay7_d280ay7_d281ay7_d282ay7_d283ay7_d284ay7_d285ay7_d286ay7_d287ay7_d287ay7_d288ay7_d289ay7_d289ay7_d290ay7_d291ay7_d292ay7_d293ay7_d294ay7_d295ay7_d296ay7_d297ay7_d297ay7_d298ay7_d299ay7_d299ay7_d300ay7_d301ay7_d302ay7_d303ay7_d304ay7_d305ay7_d306ay7_d307ay7_d308ay7_d309ay7_d309ay7_d310ay7_d311ay7_d312ay7_d313ay7_d314ay7_d315ay7_d316ay7_d317ay7_d317ay7_d318ay7_d319ay7_d319ay7_d320ay7_d321ay7_d322ay7_d323ay7_d324ay7_d325ay7_d326ay7_d327ay7_d327ay7_d328ay7_d329ay7_d329ay7_d330ay7_d331ay7_d332ay7_d333ay7_d334ay7_d335ay7_d336ay7_d337ay7_d337ay7_d338ay7_d339ay7_d339ay7_d340ay7_d341ay7_d342ay7_d343ay7_d344ay7_d345ay7_d346ay7_d347ay7_d347ay7_d348ay7_d349ay7_d349ay7_d350ay7_d351ay7_d352ay7_d353ay7_d354ay7_d355ay7_d356ay7_d357ay7_d357ay7_d358ay7_d359ay7_d359ay7_d360ay7_d361ay7_d362ay7_d363ay7_d364ay7_d365ay7_d366ay7_d367ay7_d367ay7_d368ay7_d369ay7_d369ay7_d370ay7_d371ay7_d372ay7_d373ay7_d374ay7_d375ay7_d376ay7_d377ay7_d377ay7_d378ay7_d379ay7_d379ay7_d380ay7_d381ay7_d382ay7_d383ay7_d384ay7_d385ay7_d386ay7_d387ay7_d387ay7_d388ay7_d389ay7_d389ay7_d390ay7_d391ay7_d392ay7_d393ay7_d394ay7_d395ay7_d396ay7_d397ay7_d397ay7_d398ay7_d399ay7_d399ay7_d400ay7_d401ay7_d402ay7_d403ay7_d404ay7_d405ay7_d406ay7_d407ay7_d408ay7_d409ay7_d409ay7_d410ay7_d411ay7_d412ay7_d413ay7_d414ay7_d415ay7_d416ay7_d417ay7_d417ay7_d418ay7_d419ay7_d419ay7_d420ay7_d421ay7_d422ay7_d423ay7_d424ay7_d425ay7_d426ay7_d427ay7_d427ay7_d428ay7_d429ay7_d429ay7_d430ay7_d431ay7_d432ay7_d433ay7_d434ay7_d435ay7_d436ay7_d437ay7_d437ay7_d438ay7_d439ay7_d439ay7_d440ay7_d441ay7_d442ay7_d443ay7_d444ay7_d445ay7_d446ay7_d447ay7_d447ay7_d448ay7_d449ay7_d449ay7_d450ay7_d451ay7_d452ay7_d453ay7_d454ay7_d455ay7_d456ay7_d457ay7_d457ay7_d458ay7_d459ay7_d459ay7_d460ay7_d461ay7_d462ay7_d463ay7_d464ay7_d465ay7_d466ay7_d467ay7_d467ay7_d468ay7_d469ay7_d469ay7_d470ay7_d471ay7_d472ay7_d473ay7_d474ay7_d475ay7_d476ay7_d477ay7_d477ay7_d478ay7_d479ay7_d479ay7_d480ay7_d481ay7_d482ay7_d483ay7_d484ay7_d485ay7_d486ay7_d487ay7_d487ay7_d488ay7_d489ay7_d489ay7_d490ay7_d491ay7_d492ay7_d493ay7_d494ay7_d495ay7_d496ay7_d497ay7_d497ay7_d498ay7_d499ay7_d499ay7_d500ay7_d501ay7_d502ay7_d503ay7_d504ay7_d505ay7_d506ay7_d507ay7_d508ay7_d509ay7_d509ay7_d510ay7_d511ay7_d512ay7_d513ay7_d514ay7_d515ay7_d516ay7_d517ay7_d517ay7_d518ay7_d519ay7_d519ay7_d520ay7_d521ay7_d522ay7_d523ay7_d524ay7_d525ay7_d526ay7_d527ay7_d527ay7_d528ay7_d529ay7_d529ay7_d530ay7_d531ay7_d532ay7_d533ay7_d534ay7_d535ay7_d536ay7_d537ay7_d537ay7_d538ay7_d539ay7_d539ay7_d540ay7_d541ay7_d542ay7_d543ay7_d544ay7_d545ay7_d546ay7_d547ay7_d547ay7_d548ay7_d549ay7_d549ay7_d550ay7_d551ay7_d552ay7_d553ay7_d554ay7_d555ay7_d556ay7_d557ay7_d557ay7_d558ay7_d559ay7_d559ay7_d560ay7_d561ay7_d562ay7_d563ay7_d564ay7_d565ay7_d566ay7_d567ay7_d567ay7_d568ay7_d569ay7_d569ay7_d570ay7_d571ay7_d572ay7_d573ay7_d574ay7_d575ay7_d576ay7_d577ay7_d577ay7_d578ay7_d579ay7_d579ay7_d580ay7_d581ay7_d582ay7_d583ay7_d584ay7_d585ay7_d586ay7_d587ay7_d587ay7_d588ay7_d589ay7_d589ay7_d590ay7_d591ay7_d592ay7_d593ay7_d594ay7_d595ay7_d596ay7_d597ay7_d597ay7_d598ay7_d599ay7_d599ay7_d600ay7_d601ay7_d602ay7_d603ay7_d604ay7_d605ay7_d606ay7_d607ay7_d608ay7_d609ay7_d609ay7_d610ay7_d611ay7_d612ay7_d613ay7_d614ay7_d615ay7_d616ay7_d617ay7_d617ay7_d618ay7_d619ay7_d619ay7_d620ay7_d621ay7_d622ay7_d623ay7_d624ay7_d625ay7_d626ay7_d627ay7_d627ay7_d628ay7_d629ay7_d629ay7_d630ay7_d631ay7_d632ay7_d633ay7_d634ay7_d635ay7_d636ay7_d637ay7_d637ay7_d638ay7_d639ay7_d639ay7_d640ay7_d641ay7_d642ay7_d643ay7_d644ay7_d645ay7_d646ay7_d647ay7_d647ay7_d648ay7_d649ay7_d649ay7_d650ay7_d651ay7_d652ay7_d653ay7_d654ay7_d655ay7_d656ay7_d657ay7_d657ay7_d658ay7_d659ay7_d659ay7_d660ay7_d661ay7_d662ay7_d663ay7_d664ay7_d665ay7_d666ay7_d667ay7_d667ay7_d668ay7_d669ay7_d669ay7_d670ay7_d671ay7_d672ay7_d673ay7_d674ay7_d675ay7_d676ay7_d677ay7_d677ay7_d678ay7_d679ay7_d679ay7_d680ay7_d681ay7_d682ay7_d683ay7_d684ay7_d685ay7_d686ay7_d687ay7_d687ay7_d688ay7_d689ay7_d689ay7_d690ay7_d691ay7_d692ay7_d693ay7_d694ay7_d695ay7_d696ay7_d697ay7_d697ay7_d698ay7_d699ay7_d699ay7_d700ay7_d701ay7_d702ay7_d703ay7_d704ay7_d705ay7_d706ay7_d707ay7_d708ay7_d709ay7_d709ay7_d710ay7_d711ay7_d712ay7_d713ay7_d714ay7_d715ay7_d716ay7_d717ay7_d717ay7_d718ay7_d719ay7_d719ay7_d720ay7_d721ay7_d722ay7_d723ay7_d724ay7_d725ay7_d726ay7_d727ay7_d727ay7_d728ay7_d729ay7_d729ay7_d730ay7_d731ay7_d732ay7_d733ay7_d734ay7_d735ay7_d736ay7_d737ay7_d737ay7_d738ay7_d739ay7_d739ay7_d740ay7_d741ay7_d742ay7_d743ay7_d744ay7_d745ay7_d746ay7_d747ay7_d747ay7_d748ay7_d749ay7_d749ay7_d750ay7_d751ay7_d752ay7_d753ay7_d754ay7_d755ay7_d756ay7_d757ay7_d757ay7_d758ay7_d759ay7_d759ay7_d760ay7_d761ay7_d762ay7_d763ay7_d764ay7_d765ay7_d766ay7_d767ay7_d767ay7_d768ay7_d769ay7_d769ay7_d770ay7_d771ay7_d772ay7_d773ay7_d774ay7_d775ay7_d776ay7_d777ay7_d777ay7_d778ay7_d779ay7_d779ay7_d780ay7_d781ay7_d782ay7_d783ay7_d784ay7_d785ay7_d786ay7_d787ay7_d787ay7_d788ay7_d789ay7_d789ay7_d790ay7_d791ay7_d792ay7_d793ay7_d794ay7_d795ay7_d796ay7_d797ay7_d797ay7_d798ay7_d799ay7_d799ay7_d800ay7_d801ay7_d802ay7_d803ay7_d804ay7_d805ay7_d806ay7_d807ay7_d808ay7_d809ay7_d809ay7_d810ay7_d811ay7_d812ay7_d813ay7_d814ay7_d815ay7_d816ay7_d817ay7_d817ay7_d818ay7_d819ay7_d819ay7_d820ay7_d821ay7_d822ay7_d823ay7_d824ay7_d825ay7_d826ay7_d827ay7_d827ay7_d828ay7_d829ay7_d829ay7_d830ay7_d831ay7_d832ay7_d833ay7_d834ay7_d835ay7_d836ay7_d837ay7_d837ay7_d838ay7_d839ay7_d839ay7_d840ay7_d841ay7_d842ay7_d843ay7_d844ay7_d845ay7_d846ay7_d847ay7_d847ay7_d848ay7_d849ay7_d849ay7_d850ay7_d851ay7_d852ay7_d853ay7_d854ay7_d855ay7_d856ay7_d857ay7_d857ay7_d858ay7_d859ay7_d859ay7_d860ay7_d861ay7_d862ay7_d863ay7_d864ay7_d865ay7_d866ay7_d867ay7_d867ay7_d868ay7_d869ay7_d869ay7_d870ay7_d871ay7_d872ay7_d873ay7_d874ay7_d875ay7_d876ay7_d877ay7_d877ay7_d878ay7_d879ay7_d879ay7_d880ay7_d881ay7_d882ay7_d883ay7_d884ay7_d885ay7_d886ay7_d887ay7_d887ay7_d888ay7_d889ay7_d889ay7_d890ay7_d891ay7_d892ay7_d893ay7_d894ay7_d895ay7_d896ay7_d897ay7_d897ay7_d898ay7_d899ay7_d899ay7_d900ay7_d901ay7_d902ay7_d903ay7_d904ay7_d905ay7_d906ay7_d907ay7_d908ay7_d909ay7_d909ay7_d910ay7_d911ay7_d912ay7_d913ay7_d914ay7_d915ay7_d916ay7_d917ay7_d917ay7_d918ay7_d919ay7_d919ay7_d920ay7_d921ay7_d922ay7_d923ay7_d924ay7_d925ay7_d926ay7_d927ay7_d927ay7_d928ay7_d929ay7_d929ay7_d930ay7_d931ay7_d932ay7_d933ay7_d934ay7_d935ay7_d936ay7_d937ay7_d937ay7_d938ay7_d939ay7_d939ay7_d940ay7_d941ay7_d942ay7_d943ay7_d944ay7_d945ay7_d946ay7_d947ay7_d947ay7_d948ay7_d949ay7_d949ay7_d950ay7_d951ay7_d952ay7_d953ay7_d954ay7_d955ay7_d956ay7_d957ay7_d957ay7_d958ay7_d959ay7_d959ay7_d960ay7_d961ay7_d962ay7_d963ay7_d964ay7_d965ay7_d966ay7_d967ay7_d967ay7_d968ay7_d969ay7_d969ay7_d970ay7_d971ay7_d972ay7_d973ay7_d974ay7_d975ay7_d976ay7_d977ay7_d977ay7_d978ay7_d979ay7_d979ay7_d980ay7_d981ay7_d982ay7_d983ay7_d984ay7_d985ay7_d986ay7_d987ay7_d987ay7_d988ay7_d989ay7_d989ay7_d990ay7_d991ay7_d992ay7_d993ay7_d994ay7_d995ay7_d996ay7_d997ay7_d997ay7_d998ay7_d999ay7_d999ay7_d1000ay7_d1001ay7_d1002ay7_d1003ay7_d1004ay7_d1005ay7_d1006ay7_d1007ay7_d1008ay7_d1009ay7_d1009ay7_d1010ay7_d1011ay7_d1012ay7_d1013ay7_d1014ay7_d1015ay7_d1016ay7_d1017ay7_d1017ay7_d1018ay7_d1019ay7_d1019ay7_d1020ay7_d1021ay7_d1022ay7_d1023ay7_d1024ay7_d1025ay7_d1026ay7_d1027ay7_d1027ay7_d1028ay7_d1029ay7_d1029ay7_d1030ay7_d1031ay7_d1032ay7_d1033ay7_d1034ay7_d1035ay7_d1036ay7_d1037ay7_d1037ay7_d1038ay7_d1039ay7_d1039ay7_d1040ay7_d1041ay7_d1042ay7_d1043ay7_d1044ay7_d1045ay7_d1046ay7_d1047ay7_d1047ay7_d1048ay7_d1049ay7_d1049ay7_d1050ay7_d1051ay7_d1052ay7_d1053ay7_d1054ay7_d1055ay7_d1056ay7_d1057ay7_d1057ay7_d1058ay7_d1059ay7_d1059ay7_d1060ay7_d1061ay7_d1062ay7_d1063ay7_d1064ay7_d1065ay7_d1066ay7_d1067ay7_d1067ay7_d1068ay7_d1069ay7_d1069ay7_d1070ay7_d1071ay7_d1072ay7_d1073ay7_d1074ay7_d1075ay7_d1076ay7_d1077ay7_d1077ay7_d1078ay7_d1079ay7_d1079ay7_d1080ay7_d1081ay7_d1082ay7_d1083ay7_d1084ay7_d1085ay7_d1086ay7_d1087ay7_d1087ay7_d1088ay7_d1089ay7_d1089ay7_d1090ay7_d1091ay7_d1092ay7_d1093ay7_d1094ay7_d1095ay7_d1096ay7_d1097ay7_d1097ay7_d1098ay7_d1099ay7_d1099ay7_d1100ay7_d1101ay7_d1102ay7_d1103ay7_d1104ay7_d1105ay7_d1106ay7_d1107ay7_d1108ay7_d1109ay7_d1109ay7_d1110ay7_d1111ay7_d1112ay7_d1113ay7_d1114ay7_d1115ay7_d1116ay7_d1117ay7_d1117ay7_d1118ay7_d1119ay7_d1119ay7_d1120ay7_d1121ay7_d1122ay7_d1123ay7_d1124ay7_d1125ay7_d1126ay7_d1127ay7_d1127ay7_d1128ay7_d1129ay7_d1129ay7_d1130ay7_d1131ay7_d1132ay7_d1133ay7_d1134ay7_d1135ay7_d1136ay7_d1137ay7_d1137ay7_d1138ay7_d1139ay7_d1139ay7_d1140ay7_d1141ay7_d1142ay7_d1143ay7_d1144ay7_d1145ay7_d1146ay7_d1147ay7_d1147ay7_d1148ay7_d1149ay7_d1149ay7_d1150ay7_d1151ay7_d1152ay7_d1153ay7_d1154ay7_d1155ay7_d1156ay7_d1157ay7_d1157ay7_d1158ay7_d1159ay7_d1159ay7_d1160ay7_d1161ay7_d1162ay7_d1163ay7_d1164ay7_d1165ay7_d1166ay7_d1167ay7_d1167ay7_d1168ay7_d1169ay7_d1169ay7_d1170ay7_d1171ay7_d1172ay7_d1173ay7_d1174ay7_d1175ay7_d1176ay7_d1177ay7_d1177ay7_d1178ay7_d1179ay7_d1179ay7_d1180ay7_d1181ay7_d1182ay7_d1183ay7_d1184ay7_d1185ay7_d1186ay7_d1187ay7_d1187ay7_d1188ay7_d1189ay7_d1189ay7_d1190ay7_d1191ay7_d1192ay7_d1193ay7_d1194ay7_d1195ay7_d1196ay7_d1197ay7_d1197ay7_d1198ay7_d1199ay7_d1199ay7_d1200ay7_d1201ay7_d1202ay7_d1203ay7_d1204ay7_d1205ay7_d1206ay7_d1207ay7_d1208ay7_d1209ay7_d1209ay7_d1210ay7_d1211ay7_d1212ay7_d1213ay7_d1214ay7_d1215ay7_d1216ay7_d1217ay7_d1217ay7_d1218ay7_d1219ay7_d1219ay7_d1220ay7_d1221ay7_d1222ay7_d1223ay7_d1224ay7_d1225ay7_d1226ay7_d1227ay7_d1227ay7_d1228ay7_d1229ay7_d1229ay7_d1230ay7_d1231ay7_d1232ay7_d1233ay7_d1234ay7_d1235ay7_d1236ay7_d1237ay7_d1237ay7_d1238ay7_d1239ay7_d1239ay7_d1240ay7_d1241ay7_d1242ay7_d1243ay7_d1244ay7_d1245ay7_d1246ay7_d1247ay7_d1247ay7_d1248ay7_d1249ay7_d1249ay7_d1250ay7_d1251ay7_d1252ay7_d1253ay7_d1254ay7_d1255ay7_d1256ay7_d1257ay7_d1257ay7_d1258ay7_d1259ay7_d1259ay7_d1260ay7_d1261ay7_d1262ay7_d1263ay7_d1264ay7_d1265ay7_d1266ay7_d1267ay7_d1267ay7_d1268ay7_d1269ay7_d1269ay7_d1270ay7_d1271ay7_d1272ay7_d1273ay7_d1274ay7_d1275ay7_d1276ay7_d1277ay7_d1277ay7_d1278ay7_d1279ay7_d1279ay7_d1280ay7_d1281ay7_d1282ay7_d1283ay7_d1284ay7_d1285ay7_d1286ay7_d1287ay7_d1287ay7_d1288ay7_d1289ay7_d1289ay7_d1290ay7_d1291ay7_d1292ay7_d1293ay7_d1294ay7_d1295ay7_d1296ay7_d1297ay7_d1297ay7_d1298ay7_d1299ay7_d1299ay7_d1300ay7_d1301ay7_d1302ay7_d1303ay7_d1304ay7_d1305ay7_d1306ay7_d1307ay7_d1308ay7_d1309ay7_d1309ay7_d1310ay7_d1311ay7_d1312ay7_d1313ay7_d1314ay7_d1315ay7_d1316ay7_d1317ay7_d1317ay7_d1318ay7_d1319ay7_d1319ay7_d1320ay7_d1321ay7_d1322ay7_d1323ay7_d1324ay7_d1325ay7_d1326ay7_d1327ay7_d1327ay7_d1328ay7_d1329ay7_d1329ay7_d1330ay7_d1331ay7_d1332ay7_d1333ay7_d1334ay7_d1335ay7_d1336ay7_d1337ay7_d1337ay7_d1338ay7_d1339ay7_d1339ay7_d1340ay7_d1341ay7_d1342ay7_d1343ay7_d1344ay7_d1345ay7_d1346ay7_d1347ay7_d1347ay7_d1348ay7_d1349ay7_d1349ay7_d1350ay7_d1351ay7_d1352ay7_d1353ay7_d1354ay7_d1355ay7_d1356ay7_d1357ay7_d1357ay7_d1358ay7_d1359ay7_d1359ay7_d1360ay7_d1361ay7_d1362ay7_d1363ay7_d1364ay7_d1365ay7_d1366ay7_d1367ay7_d1367ay7_d1368ay7_d1369ay7_d1369ay7_d1370ay7_d1371ay7_d1372ay7_d1373ay7_d1374ay7_d1375ay7_d1376ay7_d1377ay7_d1377ay7_d1378ay7_d1379ay7_d1379ay7_d1380ay7_d1381ay7_d1382ay7_d1383ay7_d1384ay7_d1385ay7_d1386ay7_d1387ay7_d1387ay7_d1388ay7_d1389ay7_d1389ay7_d1390ay7_d1391ay7_d1392ay7_d1393ay7_d1394ay7_d1395ay7_d1396ay7_d1397ay7_d1397ay7_d1398ay7_d1399ay7_d1399ay7_d1400ay7_d1401ay7_d1402ay7_d1403ay7_d1404ay7_d1405ay7_d1406ay7_d1407ay7_d1408ay7_d1409ay7_d1409ay7_d1410ay7_d1411ay7_d1412ay7_d1413ay7_d1414ay7_d1415ay7_d1416ay7_d1417ay7_d1417ay7_d1418ay7_d1419ay7_d1419ay7_d1420ay7_d1421ay7_d1422ay7_d1423ay7_d1424ay7_d1425ay7_d1426ay7_d1427ay7_d1427ay7_d1428ay7_d1429ay7_d1429ay7_d1430ay7_d1431ay7_d1432ay7_d1433ay7_d1434ay7_d1435ay7_d1436ay7_d1437ay7_d1437ay7_d1438ay7_d1439ay7_d1439ay7_d1440ay7_d1441ay7_d1442ay7_d1443ay7_d1444ay7_d1445ay7_d1446ay7_d1447ay7_d1447ay7_d1448ay7_d1449ay7_d1449ay7_d1450ay7_d1451ay7_d1452ay7_d1453ay7_d1454ay7_d1455ay7_d1456ay7_d1457ay7_d1457ay7_d1458ay7_d1459ay7_d1459ay7_d1460ay7_d1461ay7_d1462ay7_d1463ay7_d1464ay7_d1465ay7_d1466ay7_d1467ay7_d1467ay7_d1468ay7_d1469ay7_d1469ay7_d1470ay7_d1471ay7_d1472ay7_d1473ay7_d1474ay7_d1475ay7_d1476ay7_d1477ay7_d1477ay7_d1478ay7_d1479ay7_d
---

```
kable(head(fa_expr_norm), caption = "Loaded with myEasyLad() fa normalized")
```

Table 3: Loaded with myEasyLad() fa normalized

```
## Load proteome data
pfa_expr <- load_fix_row_names(file = local_pfa_file, rownames.col = 1)
kable(head(pfa_expr), caption = "Loaded with myEasyLad() pfa")
```

Table 4: Loaded with myEasyLad() pfa

```
## Load proteome data normalized
pfa_expr_norm <- load_fix_row_names(file = local_pfa_file_norm, rownames.col = 1)
kable(head(pfa_expr_norm), caption = "Loaded with myEasyLad() pfa normalized")
```

Table 5: Loaded with myEasyLad() pfa normalized

```
## Load transcriptome metadata
fa_meta <- read.delim(file = trans_metadata_file, sep = "\t", header = TRUE)
kable(fa_meta, caption = "Metadata for the transcriptome dataset fa")
```

Table 6: Metadata for the transcriptome dataset fa

dataType	sampleName	condition	sampleNumber	color
1 transcriptome	day14_1	day14	1	#FF4400
2 transcriptome	day14_2	day14	2	#FF4400
3 transcriptome	day14_3	day14	3	#FF4400
4 transcriptome	day1_1	day1	1	#BBD7FF
5 transcriptome	day1_2	day1	2	#BBD7FF
6 transcriptome	day1_3	day1	3	#BBD7FF
7 transcriptome	day2_1	day1	1	#F0BBFF
8 transcriptome	day2_2	day1	2	#F0BBFF
9 transcriptome	day2_3	day1	3	#F0BBFF
10 transcriptome	day3_1	day3	1	#FFFFDD
11 transcriptome	day3_2	day3	2	#FFFFDD
12 transcriptome	day3_3	day3	3	#FFFFDD
13 transcriptome	day7_1	day7	1	#FFDD88
14 transcriptome	day7_2	day7	2	#FFDD88
15 transcriptome	day7_3	day7	3	#FFDD88
16 transcriptome	normal_1	normal	1	#BBFFBB
17 transcriptome	normal_2	normal	2	#BBFFBB
18 transcriptome	normal_3	normal	3	#BBFFBB

```
## Load proteome metadata
pfa_meta <- read.delim(file = prot_metadata_file, sep = "\t", header = TRUE)
kable(pfa_meta, caption = "Metadata for the proteome dataset pfa")
```

Table 7: Metadata for the proteome dataset pfa

dataType	sampleName	condition	sampleNumber	color
proteome	normal_1	normal	1	#BBFFBB
proteome	normal_2	normal	2	#BBFFBB
proteome	day1_1	day1	1	#FFFFDD
proteome	day1_2	day1	2	#FFFFDD
proteome	day2_1	day2	1	#BBD7FF
proteome	day2_2	day2	1	#BBD7FF
proteome	day3_1	day3	1	#F0BBFF
proteome	day3_2	day3	2	#F0BBFF
proteome	day7_1	day7	1	#FFDD88
proteome	day7_2	day7	2	#FFDD88
proteome	day14_1	day14	1	#FF4400
proteome	day14_2	day14	2	#FF4400

Structure de chaque dataframe.

```
str(fa_expr)
```

```
'data.frame': 46679 obs. of 18 variables:
 $ day1_1 : num 2278.8 0 36.3 13.2 0 ...
 $ day1_2 : num 1786.5 0 22.15 7.15 27.9 ...
 $ day1_3 : num 2368.62 0 39.48 1.12 6.9 ...
```

```

$ day14_1 : num  627.758 0 14.471 0.867 5.692 ...
$ day14_2 : num  559.2 0 10.2 0 1.9 ...
$ day14_3 : num  611.434 0 31.691 0 0.655 ...
$ day2_1  : num  2145.22 0 300.56 1.71 57.38 ...
$ day2_2  : num  262.45 0 4.77 0 0 ...
$ day2_3  : num  745.84 0 123.9 5.26 38.9 ...
$ day3_1  : num  987.185 0 51.856 0.802 8.931 ...
$ day3_2  : num  1077.65 0 8.43 0 6.97 ...
$ day3_3  : num  1335.1 0 69.9 0 0 ...
$ day7_1  : num  1096.08 0 6.67 0 7.94 ...
$ day7_2  : num  1035.846 0 6.955 0.849 101.648 ...
$ day7_3  : num  1090.04 0 42.58 1.71 0.65 ...
$ normal_1: num  483.23 0 7.35 0.86 32.06 ...
$ normal_2: num  1842.1 0 11.2 0 10.4 ...
$ normal_3: num  475.7 0 1.03 0 0 ...

```

```
str(fa_expr_norm)
```

```

'data.frame':   46679 obs. of  18 variables:
$ day1_1  : num  2711.2 0 42.8 15.5 0 ...
$ day1_2  : num  1094.75 0 13.49 4.29 17.16 ...
$ day1_3  : num  1331.588 0 21.921 0.562 3.935 ...
$ day14_1 : num  728.67 0 16.24 1.16 6.96 ...
$ day14_2 : num  603.85 0 10.8 0 2.16 ...
$ day14_3 : num  646.12 0 33.84 0 1.06 ...
$ day2_1  : num  1185.66 0 166.38 1.11 31.51 ...
$ day2_2  : num  1239.5 0 23.7 0 0 ...
$ day2_3  : num  829.6 0 137.9 5.56 43.37 ...
$ day3_1  : num  928.11 0 48.9 0.94 8.46 ...
$ day3_2  : num  1110.91 0 8.24 0 7.21 ...
$ day3_3  : num  951.1 0 49.9 0 0 ...
$ day7_1  : num  953.72 0 6.09 0 6.96 ...
$ day7_2  : num  1008.027 0 6.811 0.973 99.246 ...
$ day7_3  : num  1001.467 0 39.507 1.838 0.919 ...
$ normal_1: num  719.03 0 10.42 1.49 47.64 ...
$ normal_2: num  1247.3 0 7.45 0 6.77 ...
$ normal_3: num  809.4 0 1.7 0 0 ...

```

```
str(fa_meta)
```

```

'data.frame':   18 obs. of  5 variables:
$ dataType    : chr  "1 transcriptome" "2 transcriptome" "3 transcriptome" "4 transcriptome" ...
$ sampleName  : chr  "day14_1" "day14_2" "day14_3" "day1_1" ...
$ condition   : chr  "day14" "day14" "day14" "day1" ...
$ sampleNumber: int  1 2 3 1 2 3 1 2 3 1 ...
$ color       : chr  "#FF4400" "#FF4400" "#FF4400" "#BBD7FF" ...

```

```
str(pfa_expr)
```

```

'data.frame':   8044 obs. of  10 variables:
$ normal_1: num  531.3 221.6 26.1 4363.1 879.3 ...
$ normal_2: num  651.7 266.4 29.1 4784.1 1065.4 ...

```

```
$ day1_1 : num 335.6 175.4 57.7 4064.5 914.3 ...
$ day1_2 : num 334.8 159.4 45.8 3917.3 928.3 ...
$ day2_1 : num 197.2 234.8 81.6 4599 1000.1 ...
$ day2_2 : num 307.2 256.9 88.9 5957 1264.3 ...
$ day7_1 : num 123.2 149.9 29.9 2806.5 738.4 ...
$ day7_2 : num 272.6 315.1 44.6 6792.1 1362.1 ...
$ day14_1: num 93.7 110.6 13.6 2022.5 466.4 ...
$ day14_2: num 196.2 126.4 27.7 3226.8 714.6 ...
```

```
str(pfa_expr_norm)
```

```
'data.frame': 8044 obs. of 10 variables:  
 $ normal_1: num 5.121 3.86 0.785 8.158 5.847 ...  
 $ normal_2: num 5.033 3.743 0.565 7.908 5.742 ...  
 $ day1_1 : num 4.51 3.57 1.97 8.1 5.95 ...  
 $ day1_2 : num 4.49 3.42 1.63 8.04 5.96 ...  
 $ day2_1 : num 3.63 3.88 2.36 8.17 5.97 ...  
 $ day2_2 : num 3.93 3.67 2.15 8.21 5.97 ...  
 $ day7_1 : num 3.53 3.81 1.49 8.04 6.11 ...  
 $ day7_2 : num 3.69 3.89 1.08 8.32 6 ...  
 $ day14_1: num 3.665 3.904 0.895 8.095 5.979 ...  
 $ day14_2: num 4.03 3.39 1.21 8.06 5.89 ...
```

```
str(pfa_meta)
```

```
'data.frame': 12 obs. of 5 variables:  
 $ dataType      : chr  "proteome" "proteome" "proteome" "proteome" ...  
 $ sampleName    : chr  "normal_1"  "normal_2"  "day1_1"   "day1_2"   ...  
 $ condition     : chr  "normal"    "normal"    "day1"     "day1"     ...  
 $ sampleNumber  : int  1 2 1 2 1 1 1 2 1 2 ...  
 $ color         : chr  "#BBFFBB"  "#BBFFBB"  "#FFFFDD"  "#FFFFDD" ...
```

Je supprime le groupe day3 du fichier metadata pfa

```
pfa_meta <- pfa_meta %>% filter(condition != "day3")
str(pfa_meta)
```

```
'data.frame': 10 obs. of  5 variables:  
 $ dataType    : chr  "proteome" "proteome" "proteome" "proteome" ...  
 $ sampleName  : chr  "normal_1" "normal_2" "day1_1" "day1_2" ...  
 $ condition   : chr  "normal" "normal" "day1" "day1" ...  
 $ sampleNumber: int  1 2 1 2 1 1 1 2 1 2  
 $ color       : chr  "#BBFFBB" "#BBFFBB" "#FFFFDD" "#FFFFDD" ...
```

Les deux fichiers `fa` ne donnent pas les observations de l'échantillon dans le même ordre:

```
fa_meta$sampleName == names(fa_expr)
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE
```

Donc je réorganise les échantillons dans l'ordre de l'expérience: condition normale, puis day 1 à 14 avec les 3 répliquats.

```
sample_order <- c(paste(rep(c("normal", "day1", "day2", "day3", "day7", "day14"), each = 3),
                        1:3, sep = "_"))

fa_expr <- fa_expr[,sample_order]
fa_meta <- fa_meta[match(sample_order, fa_meta$sampleName),]

fa_expr_norm <- fa_expr_norm[,sample_order]
fa_meta <- fa_meta[match(sample_order, fa_meta$sampleName),]
```

J'ai maintenant les deux jeux de données avec pour chaque un fichier metadata et une table de counts raw ou normalized:

- fa\_expr
- fa\_expr\_norm
- fa\_meta
- pfa\_expr
- pfa\_expr\_norm
- pfa\_meta

```
head(fa_expr)
```

	normal_1	normal_2	normal_3	day1_1	day1_2	day1_3	day2_1
ENSMUSG000000000001	483.2298191	1842.14841	475.696800	2278.80022	1786.498848	2368.618959	2145.223455
ENSMUSG000000000003	0.000000	0.00000	0.000000	0.00000	0.000000	0.000000	0.000000
ENSMUSG000000000028	7.3515305	11.19676	1.034465	36.27547	22.147861	39.484949	300.558779
ENSMUSG000000000031	0.8603067	0.00000	0.000000	13.18853	7.151932	1.115304	1.711944
ENSMUSG000000000037	32.0575944	10.42322	0.000000	0.00000	27.903214	6.897842	57.382077
ENSMUSG000000000049	27.7647071	38.42445	15.903837	30.86001	4.861367	51.466810	10.796186

```
head(fa_expr_norm)
```

	normal_1	normal_2	normal_3	day1_1	day1_2	day1_3	day2_1
ENSMUSG000000000001	719.032852	1247.299971	809.375515	2711.22417	1094.745589	1331.5880610	1185.658698
ENSMUSG000000000003	0.000000	0.000000	0.000000	0.00000	0.000000	0.0000000	0.000000
ENSMUSG000000000028	10.420766	7.448588	1.700369	42.82759	13.485108	21.9214582	166.379146
ENSMUSG000000000031	1.488681	0.000000	0.000000	15.46552	4.290716	0.5620887	1.105509
ENSMUSG000000000037	47.637787	6.771444	0.000000	0.00000	17.162865	3.9346207	31.507014
ENSMUSG000000000049	41.683064	25.731487	27.205900	36.87931	3.064797	28.6665222	6.080301

```
fa_meta
```

	dataType	sampleName	condition	sampleNumber	color
16	transcriptome	normal_1	normal	1	#BBFFBB
17	transcriptome	normal_2	normal	2	#BBFFBB
18	transcriptome	normal_3	normal	3	#BBFFBB

```

4 4 transcriptome    day1_1      day1          1 #BBD7FF
5 5 transcriptome    day1_2      day1          2 #BBD7FF
6 6 transcriptome    day1_3      day1          3 #BBD7FF
7 7 transcriptome    day2_1      day1          1 #FOBBFF
8 8 transcriptome    day2_2      day1          2 #FOBBFF
9 9 transcriptome    day2_3      day1          3 #FOBBFF
10 10 transcriptome   day3_1      day3          1 #FFFFDD
11 11 transcriptome   day3_2      day3          2 #FFFFDD
12 12 transcriptome   day3_3      day3          3 #FFFFDD
13 13 transcriptome   day7_1      day7          1 #FFDD88
14 14 transcriptome   day7_2      day7          2 #FFDD88
15 15 transcriptome   day7_3      day7          3 #FFDD88
1  1 transcriptome    day14_1     day14         1 #FF4400
2  2 transcriptome    day14_2     day14         2 #FF4400
3  3 transcriptome    day14_3     day14         3 #FF4400

```

```
head(pfa_expr)
```

	normal_1	normal_2	day1_1	day1_2	day2_1	day2_2	day7_1	day7_2	day14_1	day14_2
ENSMUSG00000037686	531.2680	651.7200	335.5910	334.8460	197.1740	307.194	123.2060	272.6190	93.1000	93.1000
ENSMUSG00000027831	221.6020	266.3590	175.4090	159.4190	234.8080	256.927	149.9380	315.0590	110.5000	110.5000
ENSMUSG00000039201	26.0723	29.1331	57.7329	45.8475	81.6009	88.870	29.8560	44.5586	13.0000	13.0000
ENSMUSG00000031095	4363.0500	4784.0800	4064.4800	3917.2900	4599.0300	5957.030	2806.5200	6792.0900	2022.0000	2022.0000
ENSMUSG00000034931	879.2790	1065.3900	914.2870	928.2760	1000.1000	1264.270	738.3520	1362.0700	466.0000	466.0000
ENSMUSG00000038208	68.2225	89.8871	57.9041	76.3510	84.8474	105.245	72.8696	138.9810	40.0000	40.0000

```
head(pfa_expr_norm)
```

	normal_1	normal_2	day1_1	day1_2	day2_1	day2_2	day7_1	day7_2	day14_1	day14_2
ENSMUSG00000037686	5.1206348	5.0327406	4.506491	4.492382	3.628542	3.932456	3.530692	3.685216	3.665305	4.000000
ENSMUSG00000027831	3.8601167	3.7429066	3.571461	3.422865	3.880248	3.674968	3.813620	3.893704	3.903714	3.000000
ENSMUSG00000039201	0.7849136	0.5645967	1.972258	1.630415	2.358439	2.146880	1.492104	1.081497	0.894504	1.000000
ENSMUSG00000031095	8.1578622	7.9080330	8.103830	8.039705	8.170497	8.208356	8.038369	8.322345	8.095098	8.000000
ENSMUSG00000034931	5.8472466	5.7415172	5.951775	5.962765	5.969607	5.972364	6.112220	6.004585	5.979016	5.000000
ENSMUSG00000038208	2.1641269	2.1791889	1.976512	2.363160	2.414548	2.390031	2.774423	2.714986	2.468220	2.000000

```
pfa_meta
```

	dataType	sampleName	condition	sampleNumber	color
1	proteome	normal_1	normal	1	#BBFFBB
2	proteome	normal_2	normal	2	#BBFFBB
3	proteome	day1_1	day1	1	#FFFFDD
4	proteome	day1_2	day1	2	#FFFFDD
5	proteome	day2_1	day2	1	#BBD7FF
6	proteome	day2_2	day2	1	#BBD7FF
9	proteome	day7_1	day7	1	#FFDD88
10	proteome	day7_2	day7	2	#FFDD88
11	proteome	day14_1	day14	1	#FF4400
12	proteome	day14_2	day14	2	#FF4400

## Analyses différentielles DESeq2

Analyse d'expression différentielle pour les données de protéomique et transcriptomique => identifier les gènes/protéines significativement différemment exprimés dans le modèle FA en comparant Day 7 à Day 0.

### Modèle fa en comparant Day 7 à normal

```
#?DESeq  
library("DESeq2")
```

Préparation objet **DESeq2**

```
# id <- pfa_expr [,1]  
# pfa_expr <- pfa_expr[,-1]  
# rownames(pfa_expr) <- make.names(id, unique = TRUE)  
  
#any (fa_DataMatrix [,-1] < 0)  
  
fa_DataMatrix <- as.matrix(fa_expr)  
  
dds <- DESeqDataSetFromMatrix(countData = round(fa_DataMatrix), colData = fa_meta, design = ~ condition)  
dds
```

```
class: DESeqDataSet  
dim: 46679 18  
metadata(1): version  
assays(1): counts  
rownames(46679): ENSMUSG000000000001 ENSMUSG000000000003 ... ENSMUSG00000109577 ENSMUSG00000109578  
rowData names(0):  
colnames(18): normal_1 normal_2 ... day14_2 day14_3  
colData names(5): dataType sampleName condition sampleNumber color
```

Run fonction DESeq2

```
dds <- DESeq(dds)
```

Table de résultats du DESeq2 entre les groupes **normal** et **day7**

```
res <- results(dds, contrast=c("condition", "normal", "day7"))  
head(res)
```

```
log2 fold change (MLE): condition normal vs day7  
Wald test p-value: condition normal vs day7  
DataFrame with 6 rows and 6 columns  
  baseMean log2FoldChange      lfcSE      stat     pvalue      padj  
  <numeric>      <numeric> <numeric> <numeric> <numeric> <numeric>  
ENSMUSG000000000001 1061.10739    -0.0926767  0.371089 -0.249742  0.802787  0.936171  
ENSMUSG000000000003    0.00000      NA        NA        NA        NA        NA  
ENSMUSG000000000028   35.89103    -1.3988787  1.037433 -1.348404  0.177528  0.509671
```

```

ENSMUSG00000000031    1.85467    -1.0447828  2.258700 -0.462559  0.643680      NA
ENSMUSG00000000037    15.74259   -0.9804598  1.674099 -0.585664  0.558101  0.828730
ENSMUSG00000000049    20.97970   0.5186876  1.122010  0.462284  0.643877  0.872279

```

Combien de gènes sont **significatifs > 0.05**

```
table(res$padj < 0.05)
```

```

FALSE  TRUE
20701 1764

```

Je classe la table par **ordre décroissant de padj**

```

orderedRes <- res[ order(res$padj), ]
head(orderedRes)

```

```
log2 fold change (MLE): condition normal vs day7
```

```
Wald test p-value: condition normal vs day7
```

```
DataFrame with 6 rows and 6 columns
```

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
ENSMUSG00000024164	14670.159	-8.19459	0.593794	-13.80039	2.53467e-43	5.69414e-39
ENSMUSG00000029304	224391.951	-5.14937	0.480753	-10.71105	9.03342e-27	9.62642e-23
ENSMUSG00000061947	621.939	-4.85093	0.454278	-10.67834	1.28552e-26	9.62642e-23
ENSMUSG00000027962	2261.581	-5.22527	0.503833	-10.37103	3.35873e-25	1.88635e-21
ENSMUSG00000003617	5729.876	-3.80553	0.383311	-9.92805	3.14337e-23	1.41232e-19
ENSMUSG00000066071	3575.949	2.88208	0.301427	9.56143	1.16138e-21	4.34841e-18

Je **normalise** la table

```

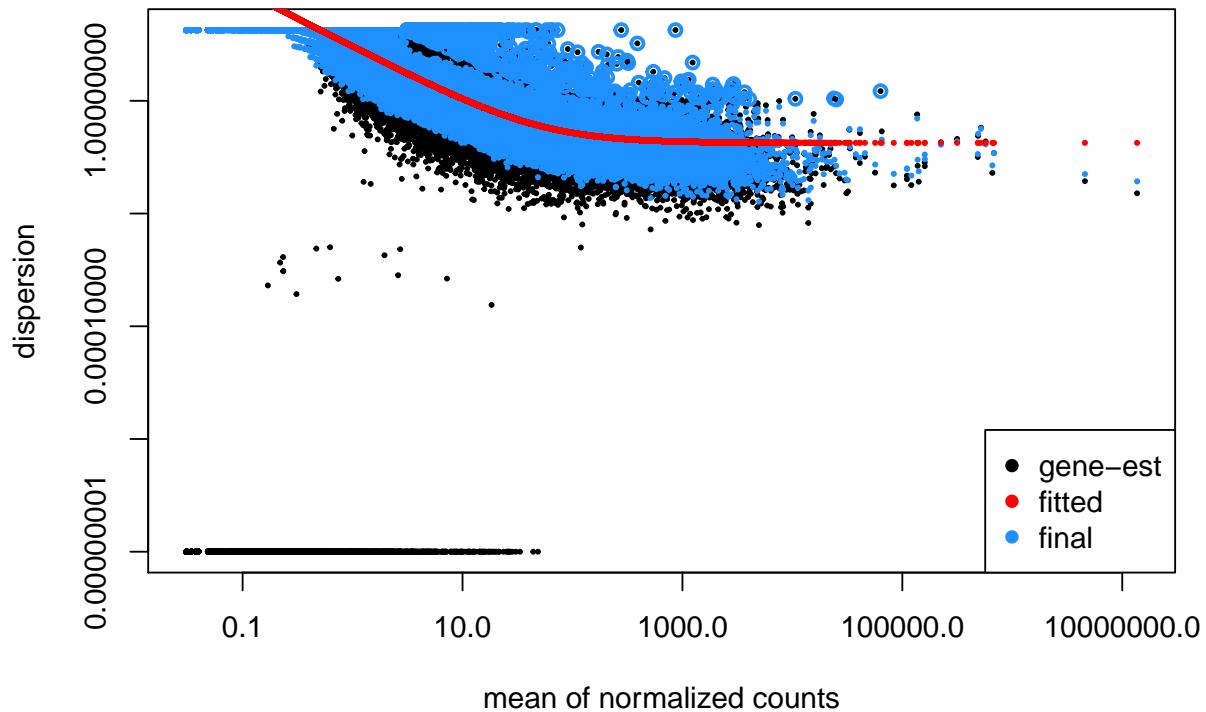
normCounts <- counts(dds, normalized = TRUE)
head(normCounts)

```

	normal_1	normal_2	normal_3	day1_1	day1_2	day1_3	day2_1	...
ENSMUSG00000000001	719.032852	1247.299971	809.375515	2711.22417	1094.745589	1331.5880610	1185.658698	1234.668621
ENSMUSG00000000003	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000000	0.000000	0.000000
ENSMUSG00000000028	10.420766	7.448588	1.700369	42.82759	13.485108	21.9214582	166.379146	183.379146
ENSMUSG00000000031	1.488681	0.000000	0.000000	15.46552	4.290716	0.5620887	1.105509	1.105509
ENSMUSG00000000037	47.637787	6.771444	0.000000	0.000000	17.162865	3.9346207	31.507014	31.507014
ENSMUSG00000000049	41.683064	25.731487	27.205900	36.87931	3.064797	28.6665222	6.080301	6.080301

Visualisation des **estimations de dispersion** de DESeq2

```
plotDispEsts(dds)
```

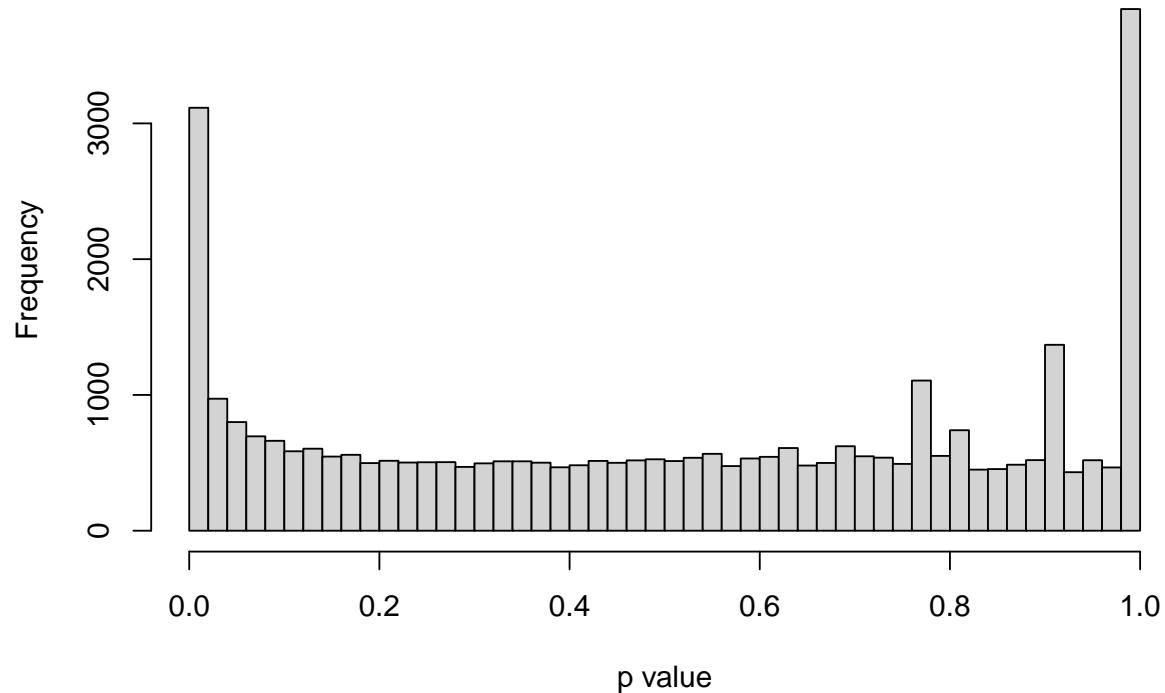


- Les points noirs sont la représentation “brute” des gènes (forte variabilité)
- La ligne de tendance rouge (dépendance des dispersions à la moyenne)
- Les points bleus représentent l'estimation de chaque gène vers la ligne rouge
- Les cercles bleus au-dessus du «nuage» principal représentent des gènes avec des valeurs aberrantes de dispersion.

Distribution des **pvalues**

```
hist(orderedRes$pvalue, breaks=0:50/50, xlab="p value", main="Histogram of nominal p values")
```

## Histogram of nominal p values



Heatmap des 20 gènes les plus différemment exprimés.

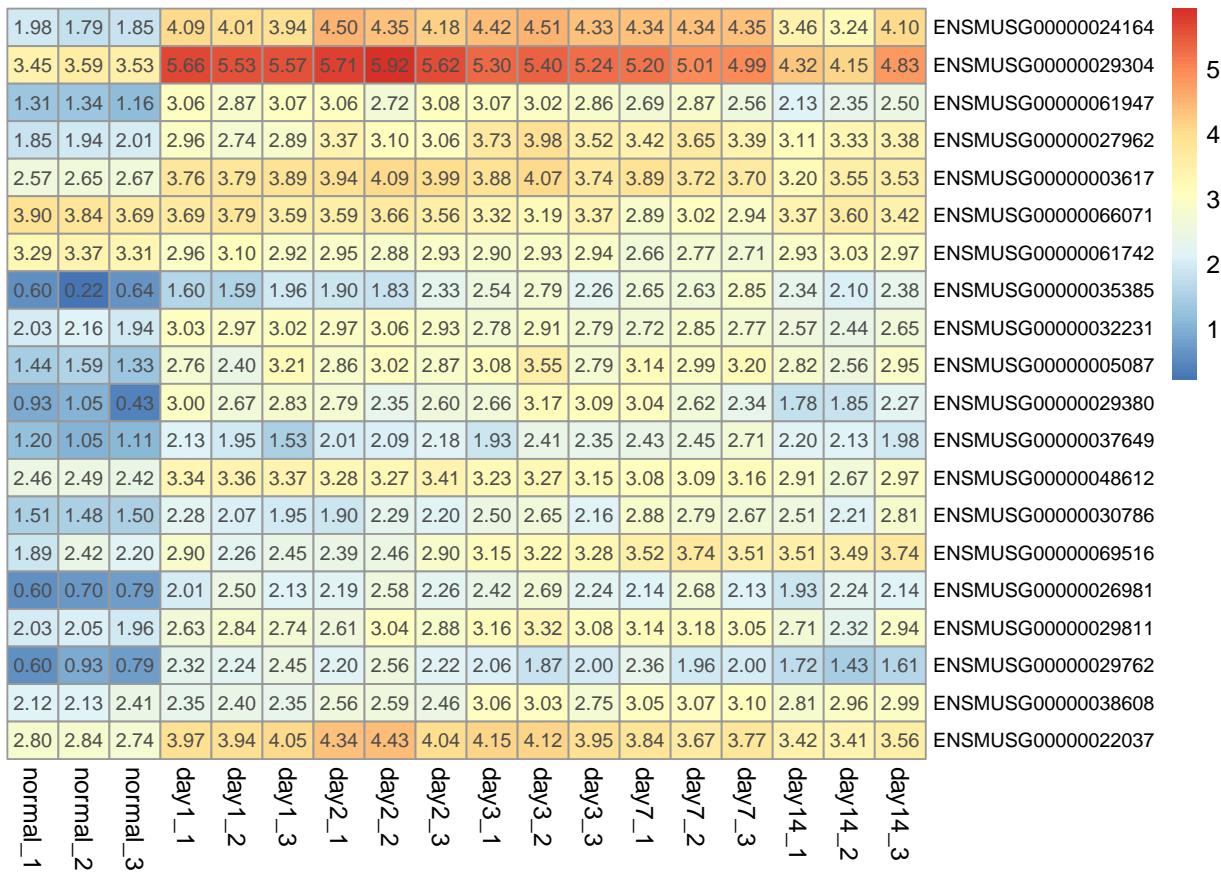
```
library(pheatmap)

# select the 20 most differentially expressed genes
select <- row.names(orderedRes[1:20, ])

# transform the counts to log10
log10_normCounts <- log10(normCounts + 1)

# get the values for the selected genes
values <- log10_normCounts[ select, ]

pheatmap(values,
         scale = "none",
         cluster_rows = FALSE,
         cluster_cols = FALSE,
         fontsize_row = 8,
         annotation_names_col = FALSE,
         #gaps_col = c(3,6),
         display_numbers = TRUE,
         number_format = "%.2f",
         height=12,
         width=6)
```



## Modèle pfa en comparant Day 7 à normal

Préparation objet DESeq2

```
# id <- pfa_expr [,1]
# pfa_expr <- pfa_expr[,-1]
# rownames(pfa_expr) <- make.names(id, unique = TRUE)

#any (fa_DataMatrix [,-1] < 0)

pfa_DataMatrix <- as.matrix(pfa_expr)

dds <- DESeqDataSetFromMatrix(countData = round(pfa_DataMatrix), colData = pfa_meta, design = ~ condition)
dds
```

```
class: DESeqDataSet
dim: 8044 10
metadata(1): version
assays(1): counts
rownames(8044): ENSMUSG00000037686 ENSMUSG00000027831 ... ENSMUSG00000027523.1 ENSMUSG00000020741.1
rowData names(0):
colnames(10): 1 2 ... 11 12
colData names(5): dataType sampleName condition sampleNumber color
```

Run fonction DESeq2

```
dds <- DESeq(dds)
```

Table de résultats du DESeq2 entre les groupes **normal** et **day7**

```
res <- results(dds, contrast=c("condition", "normal", "day7"))
head(res)
```

log2 fold change (MLE): condition normal vs day7

Wald test p-value: condition normal vs day7

DataFrame with 6 rows and 6 columns

	baseMean	log2FoldChange	lfcSE	stat	pvalue	
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<num
ENSMUSG00000037686	283.1512	1.4928568	0.191451	7.797601	0.00000000000000630949	0.00000000000000630949
ENSMUSG00000027831	193.2636	-0.0228946	0.217834	-0.105101	0.91629551187216284891	0.93704431221824
ENSMUSG00000039201	42.2126	-0.5914299	0.411140	-1.438513	0.15028861831364714874	0.21595599244640
ENSMUSG00000031095	4050.7644	-0.1184521	0.129794	-0.912619	0.36144292218839429998	0.45076695598192
ENSMUSG00000034931	900.1919	-0.2292681	0.104342	-2.197286	0.02800003925436490848	0.05041005777923
ENSMUSG00000038208	76.4645	-0.5416567	0.276803	-1.956830	0.05036751097988109716	0.08362455263217

Combien de gènes sont **significatifs**  $> 0.05$

```
table(res$padj < 0.05)
```

FALSE	TRUE
3582	4462

Je classe la table par **ordre décroissant de padj**

```
orderedRes <- res[ order(res$padj), ]
head(orderedRes)
```

log2 fold change (MLE): condition normal vs day7

Wald test p-value: condition normal vs day7

DataFrame with 6 rows and 6 columns

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
ENSMUSG00000071715	1097.288	-2.96982	0.157649	-18.8382	3.67186e-79	1.47682e-75
ENSMUSG00000024757	963.642	2.91933	0.154778	18.8614	2.36703e-79	1.47682e-75
ENSMUSG00000053303	626.809	3.02825	0.163434	18.5290	1.20598e-76	3.23363e-73
ENSMUSG00000039519	2155.892	2.77257	0.153107	18.1087	2.72021e-73	5.47034e-70
ENSMUSG00000066097	932.324	2.61823	0.145835	17.9534	4.51718e-72	7.26724e-69
ENSMUSG00000032047	13304.389	1.93889	0.109767	17.6637	7.98453e-70	1.07046e-66

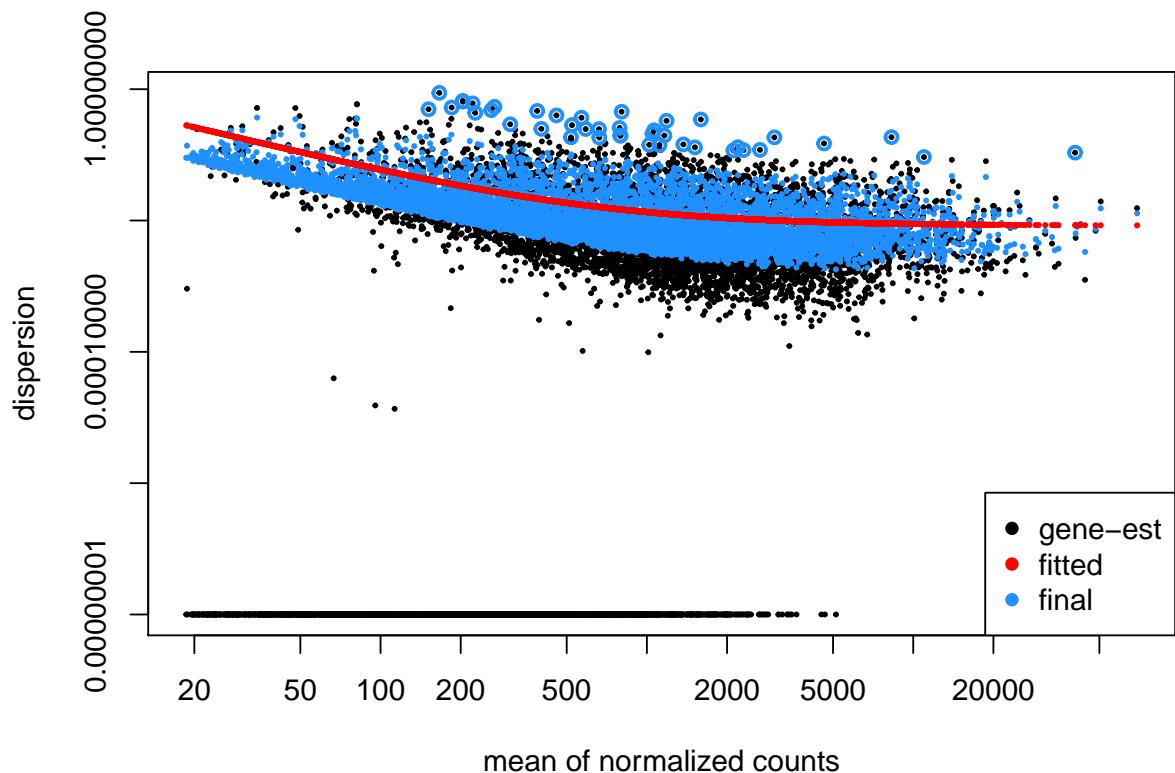
Je **normalise** la table

```
normCounts <- counts(dds, normalized = TRUE)
head(normCounts)
```

	1	2	3	4	5	6	9	...
ENSMUSG00000037686	510.05841	481.36921	327.9377	325.17673	181.56837	226.98770	165.17223	185.0408
ENSMUSG00000027831	213.24476	196.38683	170.8009	154.33761	216.59171	190.01902	201.42954	213.5081
ENSMUSG00000039201	24.97461	21.41059	56.6083	44.65113	75.57668	65.80425	40.28591	30.5011
ENSMUSG00000031095	4190.93192	3532.00967	3966.4851	3802.14103	4238.74590	4404.44867	3769.41820	4603.6561
ENSMUSG00000034931	844.33398	786.28560	892.0687	900.78807	921.66686	934.56826	991.03336	923.1714
ENSMUSG00000038208	65.31821	66.44667	56.6083	73.77144	78.34168	77.63423	98.02904	94.2156

Visualisation des **estimations de dispersion** de DESeq2

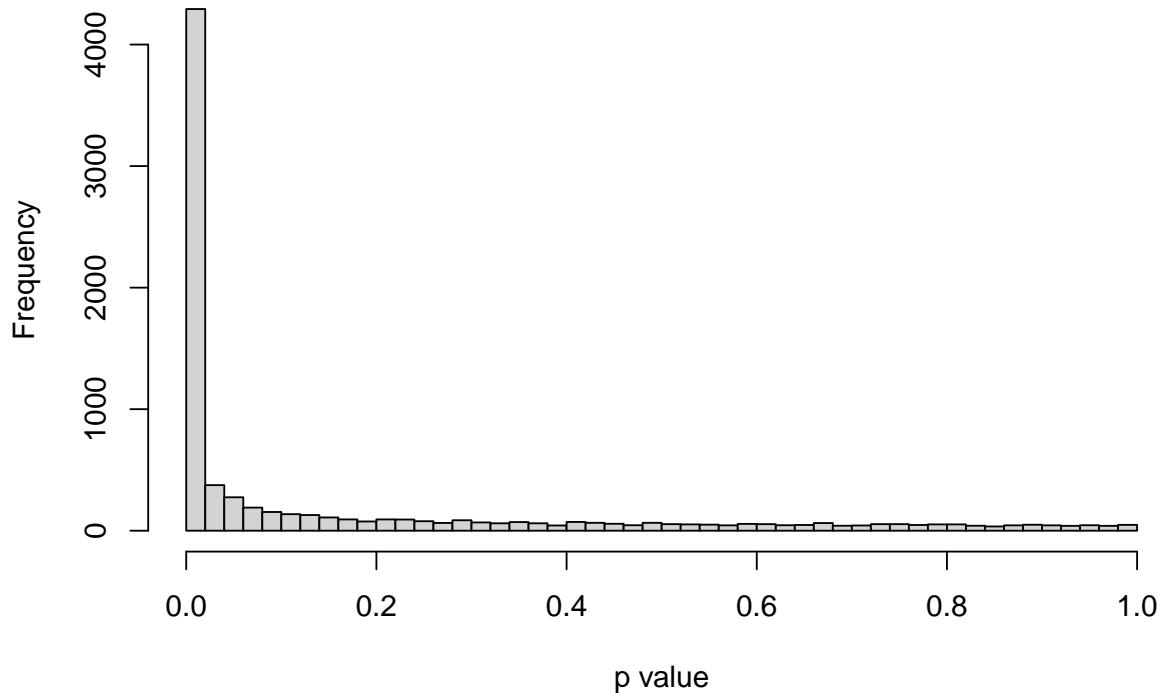
```
plotDispEsts(dds)
```



Distribution des **pvalues**

```
hist(orderedRes$pvalue, breaks=0:50/50, xlab="p value", main="Histogram of nominal p values")
```

## Histogram of nominal p values



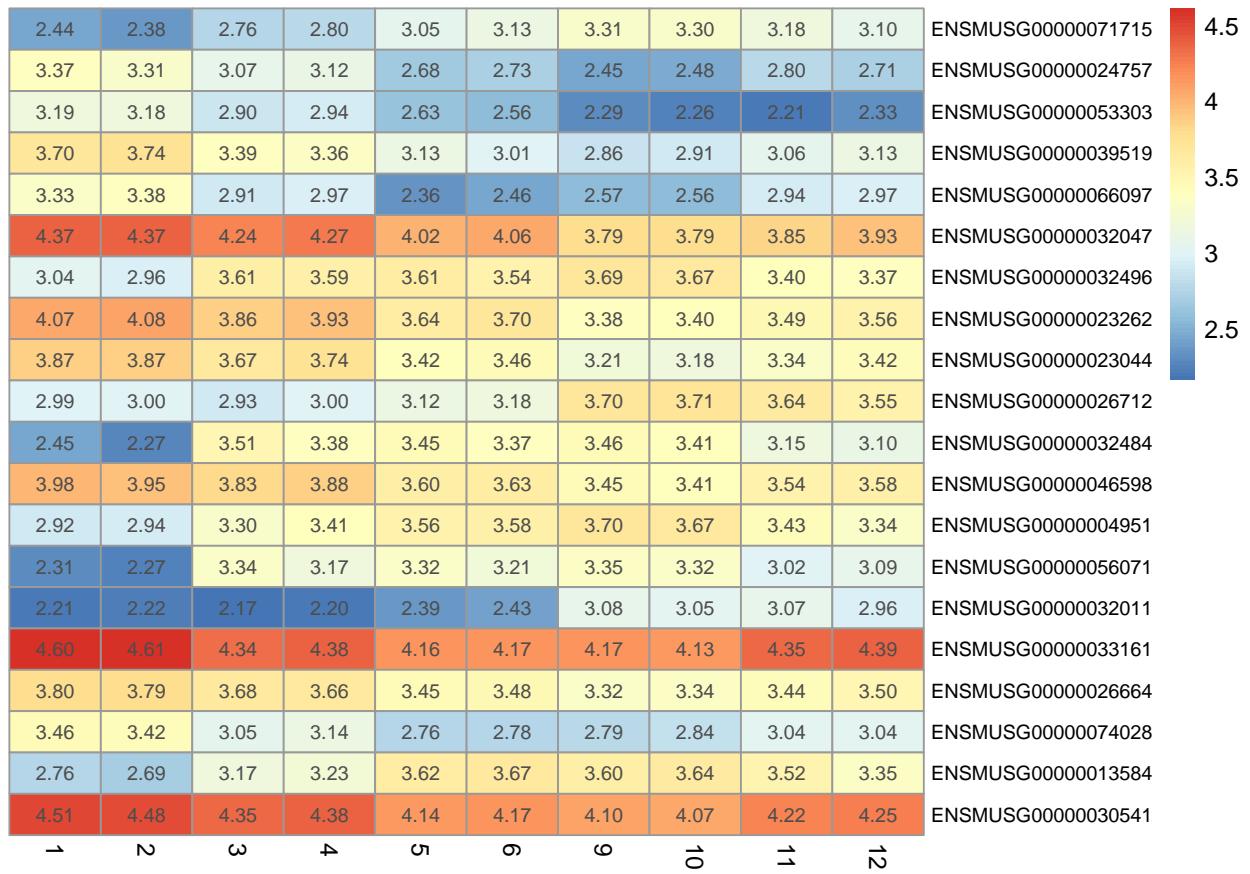
Heatmap des 20 gènes les plus différemmentiellement exprimés.

```
# select the 20 most differentially expressed genes
select <- row.names(orderedRes[1:20, ])

# transform the counts to log10
log10_normCounts <- log10(normCounts + 1)

# get the values for the selected genes
values <- log10_normCounts[ select, ]

pheatmap(values,
         scale = "none",
         cluster_rows = FALSE,
         cluster_cols = FALSE,
         fontsize_row = 8,
         annotation_names_col = FALSE,
         #gaps_col = c(3,6),
         display_numbers = TRUE,
         number_format = "%.2f",
         height=12,
         width=6)
```



## MixOmics

Analyse multi-omique (transcripto + protéo) avec, au choix, MOFA, mixOmics, mixKernel ou d'autres outils de factorisation multi-matrices. Vous pouvez soit vous focaliser sur un time point, soit intégrer les différents time points, en partant des données normalisées fournies dans le matériel supplémentaire du papier.

### Préparation des tables

Il y a 18 échantillons pour l'expérience fa et 10 pour l'expérience pfa, il faut donc supprimer les 6 échantillons supplémentaire de la table fa.

Il faut également uniformiser la table metadata en conséquence.

```
head(fa_expr_norm)
```

	normal_1	normal_2	normal_3	day1_1	day1_2	day1_3	day2_1	day2_2
ENSMUSG00000000001	719.032852	1247.299971	809.375515	2711.22417	1094.745589	1331.5880610	1185.658698	1230.6600000
ENSMUSG00000000003	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000000	0.0000000	0.0000000
ENSMUSG00000000028	10.420766	7.448588	1.700369	42.82759	13.485108	21.9214582	166.379146	170.0000000
ENSMUSG00000000031	1.488681	0.000000	0.000000	15.46552	4.290716	0.5620887	1.105509	1.1055090
ENSMUSG00000000037	47.637787	6.771444	0.000000	0.000000	17.162865	3.9346207	31.507014	31.5070140
ENSMUSG00000000049	41.683064	25.731487	27.205900	36.87931	3.064797	28.6665222	6.080301	6.0803010

```
fa <- fa_expr_norm[, - c(3, 6, 9, 10, 11, 12, 15, 18)]
head(fa)
```

	normal_1	normal_2	day1_1	day1_2	day2_1	day2_2	day7_1	
ENSMUSG000000000001	719.032852	1247.299971	2711.22417	1094.745589	1185.658698	1239.4695	953.722863	1008.0
ENSMUSG000000000003	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
ENSMUSG000000000028	10.420766	7.448588	42.82759	13.485108	166.379146	23.6540	6.091296	6.0
ENSMUSG000000000031	1.488681	0.000000	15.46552	4.290716	1.105509	0.0000	0.000000	0.0
ENSMUSG000000000037	47.637787	6.771444	0.00000	17.162865	31.507014	0.0000	6.961481	99.0
ENSMUSG000000000049	41.683064	25.731487	36.87931	3.064797	6.080301	14.1924	27.845923	6.0

```
pfa <- pfa_expr_norm
head(pfa)
```

	normal_1	normal_2	day1_1	day1_2	day2_1	day2_2	day7_1	day7_2	day14_1	
ENSMUSG00000037686	5.1206348	5.0327406	4.506491	4.492382	3.628542	3.932456	3.530692	3.685216	3.665305	4
ENSMUSG00000027831	3.8601167	3.7429066	3.571461	3.422865	3.880248	3.674968	3.813620	3.893704	3.903714	3
ENSMUSG00000039201	0.7849136	0.5645967	1.972258	1.630415	2.358439	2.146880	1.492104	1.081497	0.894504	1
ENSMUSG00000031095	8.1578622	7.9080330	8.103830	8.039705	8.170497	8.208356	8.038369	8.322345	8.095098	8
ENSMUSG00000034931	5.8472466	5.7415172	5.951775	5.962765	5.969607	5.972364	6.112220	6.004585	5.979016	5
ENSMUSG00000038208	2.1641269	2.1791889	1.976512	2.363160	2.414548	2.390031	2.774423	2.714986	2.468220	2

```
metadata <- pfa_meta[, - 1]
metadata
```

	sampleName	condition	sampleNumber	color
1	normal_1	normal	1	#BBFFBB
2	normal_2	normal	2	#BBFFBB
3	day1_1	day1	1	#FFFFDD
4	day1_2	day1	2	#FFFFDD
5	day2_1	day2	1	#BBD7FF
6	day2_2	day2	1	#BBD7FF
9	day7_1	day7	1	#FFDD88
10	day7_2	day7	2	#FFDD88
11	day14_1	day14	1	#FF4400
12	day14_2	day14	2	#FF4400

```
#Je retire le rawnames original (les chiffres 1,2,3...)
rownames(metadata) <- NULL
metadata
```

	sampleName	condition	sampleNumber	color
1	normal_1	normal	1	#BBFFBB
2	normal_2	normal	2	#BBFFBB
3	day1_1	day1	1	#FFFFDD
4	day1_2	day1	2	#FFFFDD
5	day2_1	day2	1	#BBD7FF
6	day2_2	day2	1	#BBD7FF
7	day7_1	day7	1	#FFDD88
8	day7_2	day7	2	#FFDD88
9	day14_1	day14	1	#FF4400
10	day14_2	day14	2	#FF4400

```

# add the rownames as a proper column
library(tibble)
metadata <- column_to_rownames(metadata, var = "sampleName")

rownames(metadata)

[1] "normal_1" "normal_2" "day1_1"    "day1_2"    "day2_1"    "day2_2"    "day7_1"    "day7_2"    "day14_1"

colnames(fa)

[1] "normal_1" "normal_2" "day1_1"    "day1_2"    "day2_1"    "day2_2"    "day7_1"    "day7_2"    "day14_1"

colnames(pfa)

[1] "normal_1" "normal_2" "day1_1"    "day1_2"    "day2_1"    "day2_2"    "day7_1"    "day7_2"    "day14_1"

```

## Filtres

Je regarde combien il y a de lignes avec des zéros dans la table fa et je les supprime et je filtre les genes où il y au moins de 3 échantillons avec des counts supérieurs ou égaux à 5

```

dim(fa)

[1] 46679      10

#voir le nombre de lignes avec zero counts
rs <- rowSums(fa)
nbgenes_at_zeros <- length(which(rs==0))
nbgenes_at_zeros

[1] 13269

#Je supprime les lignes avec zero counts
fa <- fa[rowSums(fa[, -1])>0, ]
#Je ne garde que les genes où il y a moins de 3 échantillons avec des counts supérieurs ou égaux à 5.
fa <- fa[rowSums((fa[, -1])>=5)>= 3 , ]
dim(fa)

[1] 22459      10

```

Idem pour la table pfa

```

dim(pfa)

[1] 8044      10

```

```
rs <- rowSums(pfa)
nbgenes_at_zeros <- length(which(rs==0))
nbgenes_at_zeros
```

```
[1] 0
```

```
pfa <- pfa[rowSums((pfa[, -1])>=5)>= 3 , ]
dim(pfa)
```

```
[1] 5277    10
```

Il faut également transposer les counts tables

```
fa_t <- t(fa)
str(fa_t)
```

```
num [1:10, 1:22459] 719 1247 2711 1095 1186 ...
- attr(*, "dimnames")=List of 2
..$ : chr [1:10] "normal_1" "normal_2" "day1_1" "day1_2" ...
..$ : chr [1:22459] "ENSMUSG00000000001" "ENSMUSG00000000028" "ENSMUSG00000000037" "ENSMUSG0000000004
```

```
dim(fa_t)
```

```
[1] 10 22459
```

```
class(fa_t)
```

```
[1] "matrix" "array"
```

```
pfa_t <- t(pfa_expr_norm)
str(pfa_t)
```

```
num [1:10, 1:8044] 5.12 5.03 4.51 4.49 3.63 ...
- attr(*, "dimnames")=List of 2
..$ : chr [1:10] "normal_1" "normal_2" "day1_1" "day1_2" ...
..$ : chr [1:8044] "ENSMUSG00000037686" "ENSMUSG00000027831" "ENSMUSG00000039201" "ENSMUSG00000031095
```

```
dim(pfa_t)
```

```
[1] 10 8044
```

```
class(pfa_t)
```

```
[1] "matrix" "array"
```

```
dim(metadata)
```

```
[1] 10 3
```

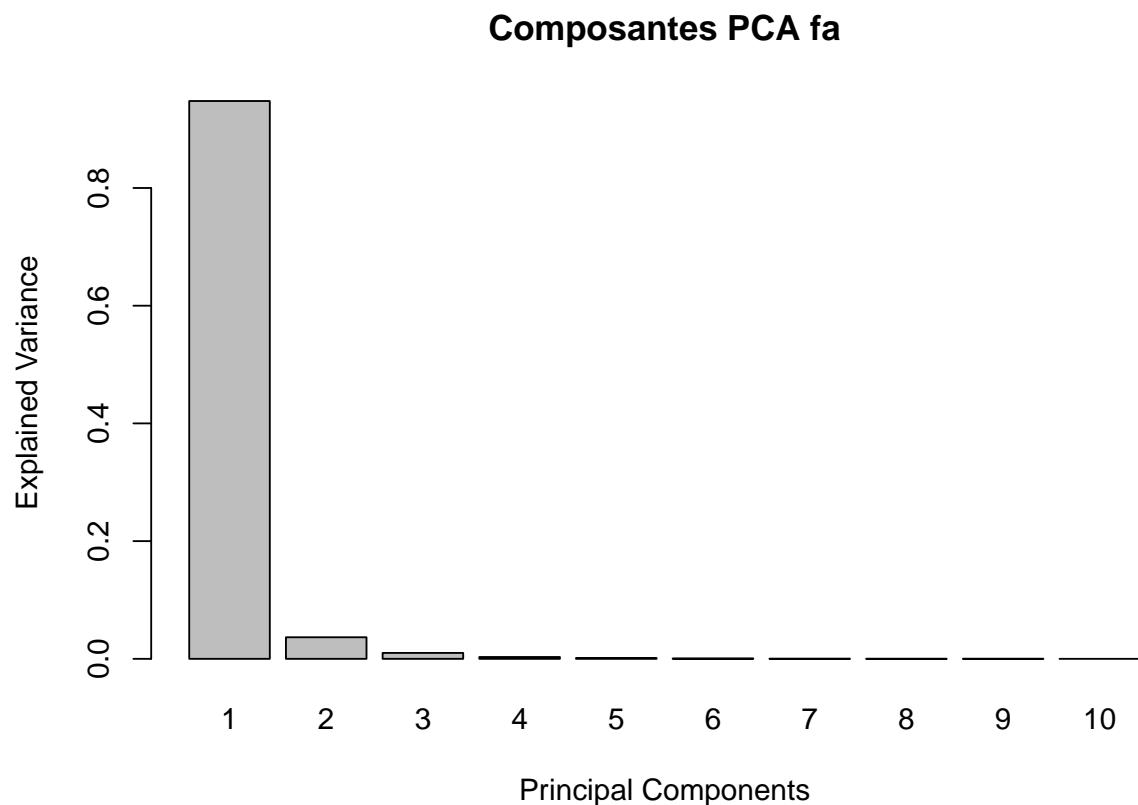
J'ai donc maintenant les deux counts tables avec 10 échantillons

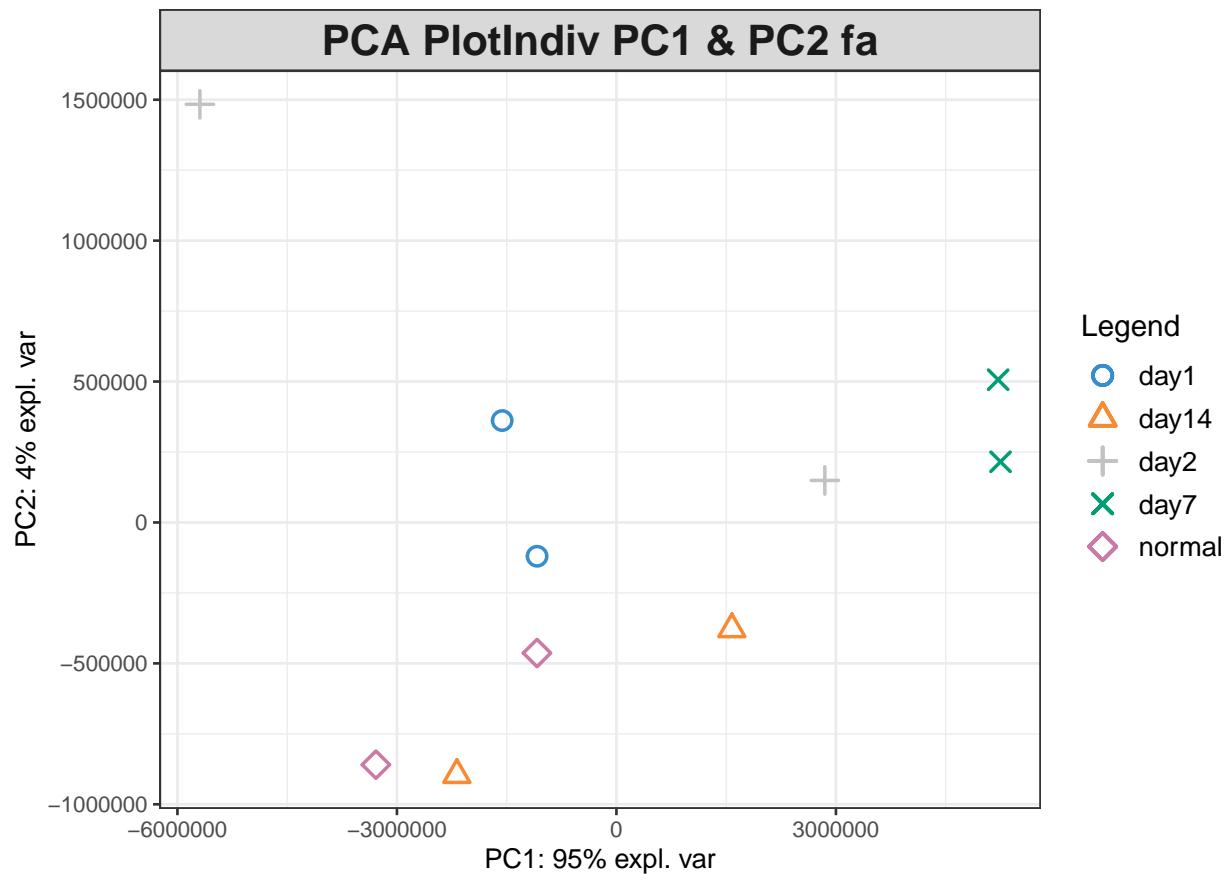
## Import package

```
library(mixOmics)
```

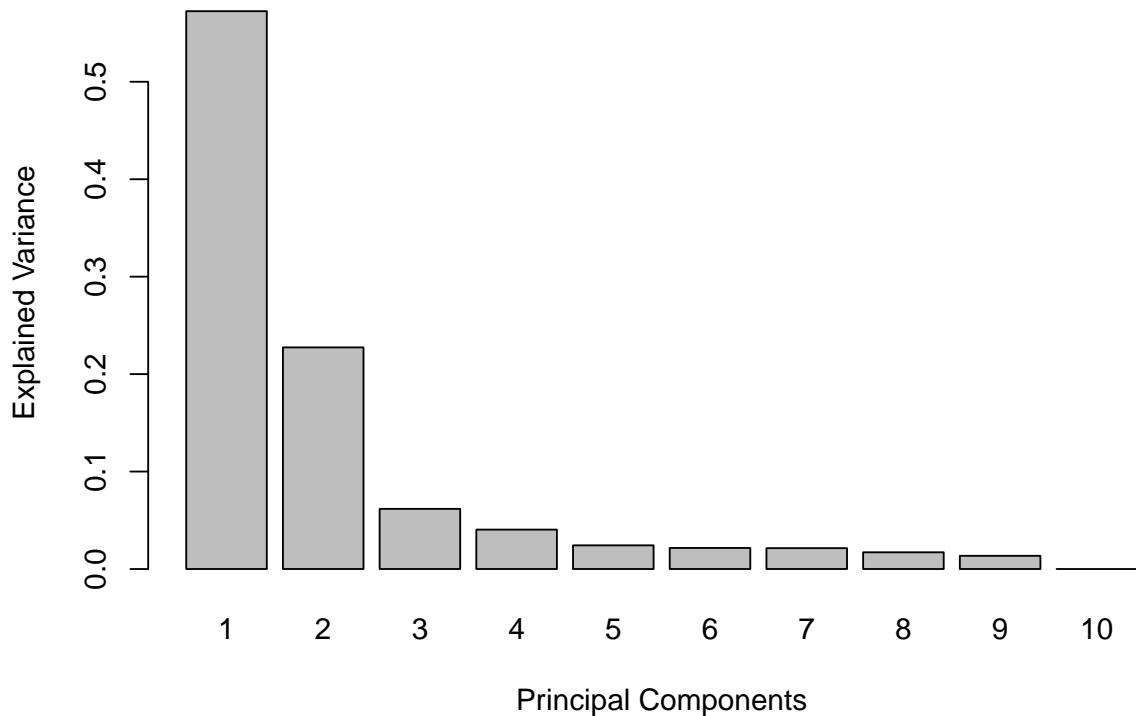
## Analyses uni omic

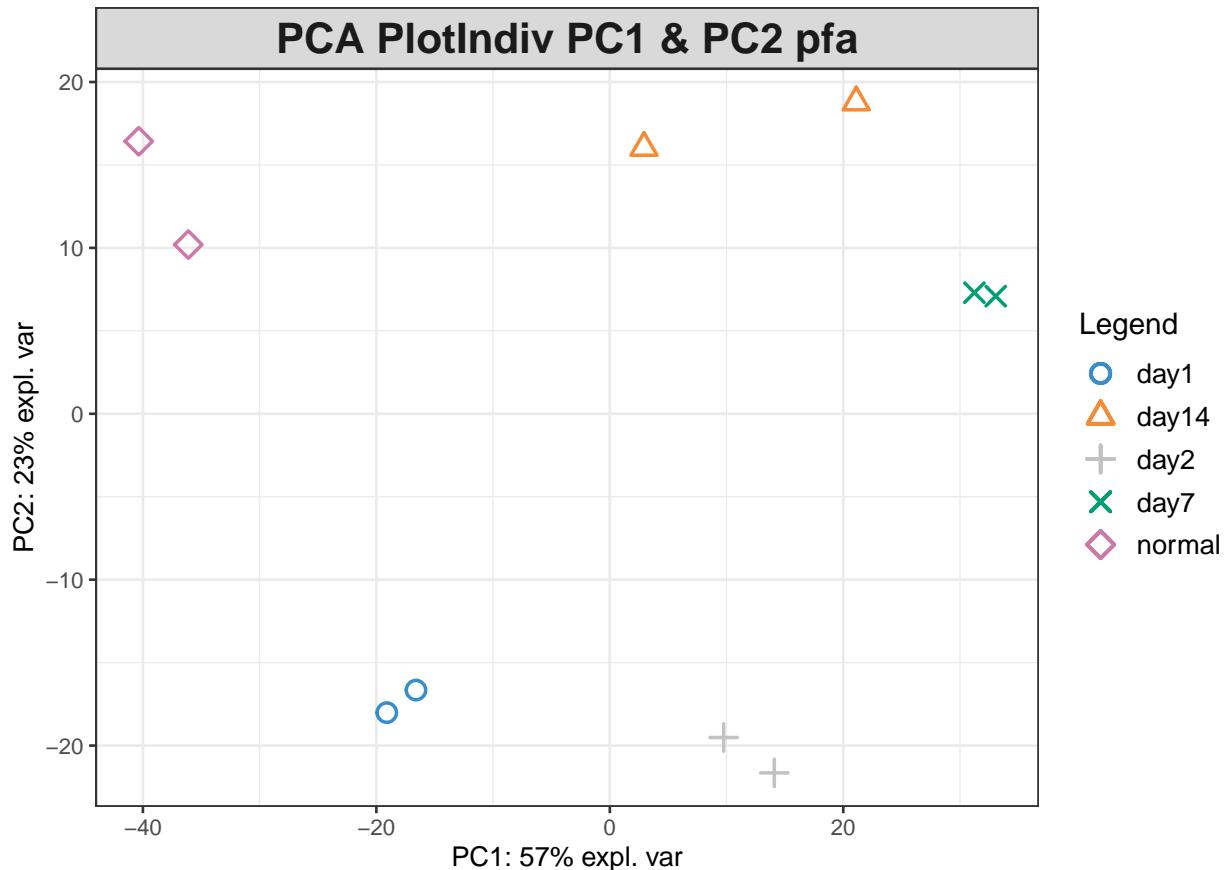
### PCA





### Composantes PCA pfa





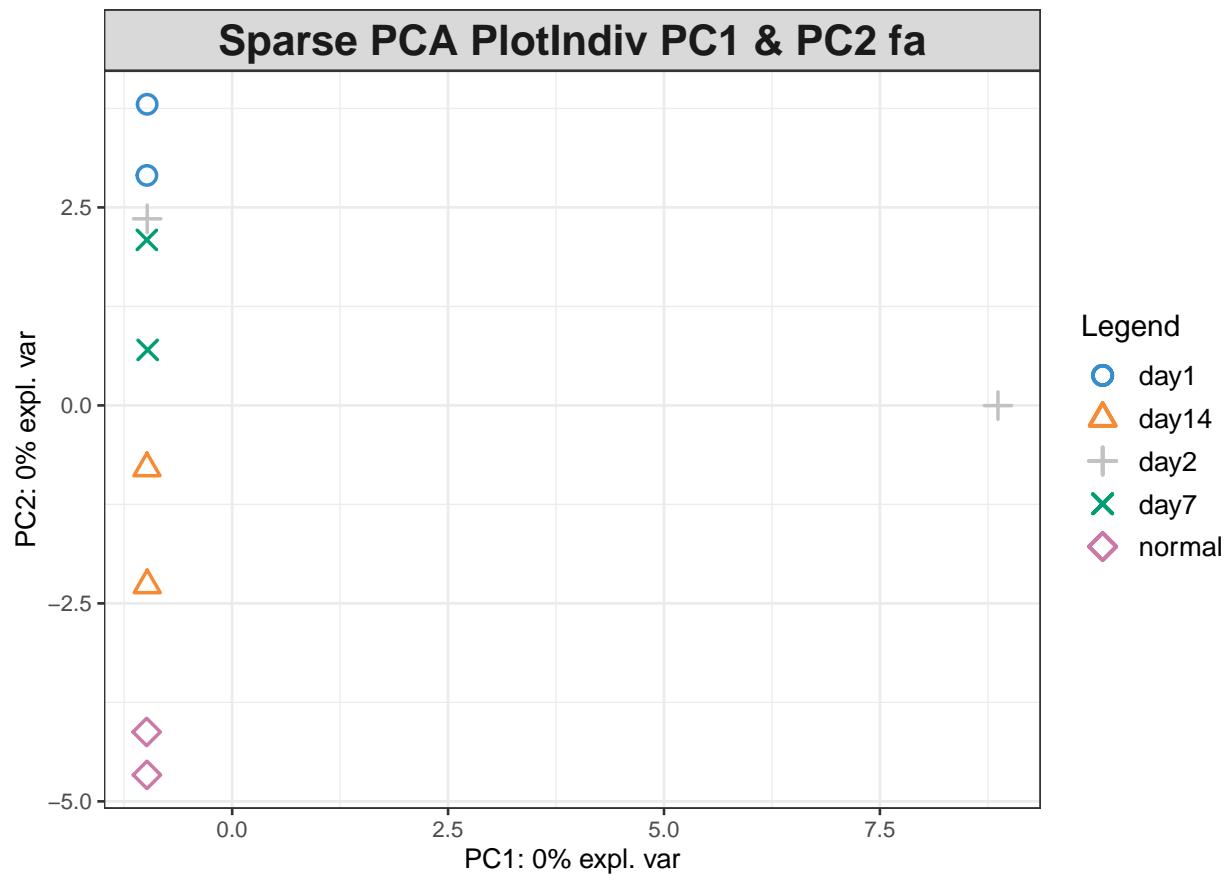
### Sparse PCA

```

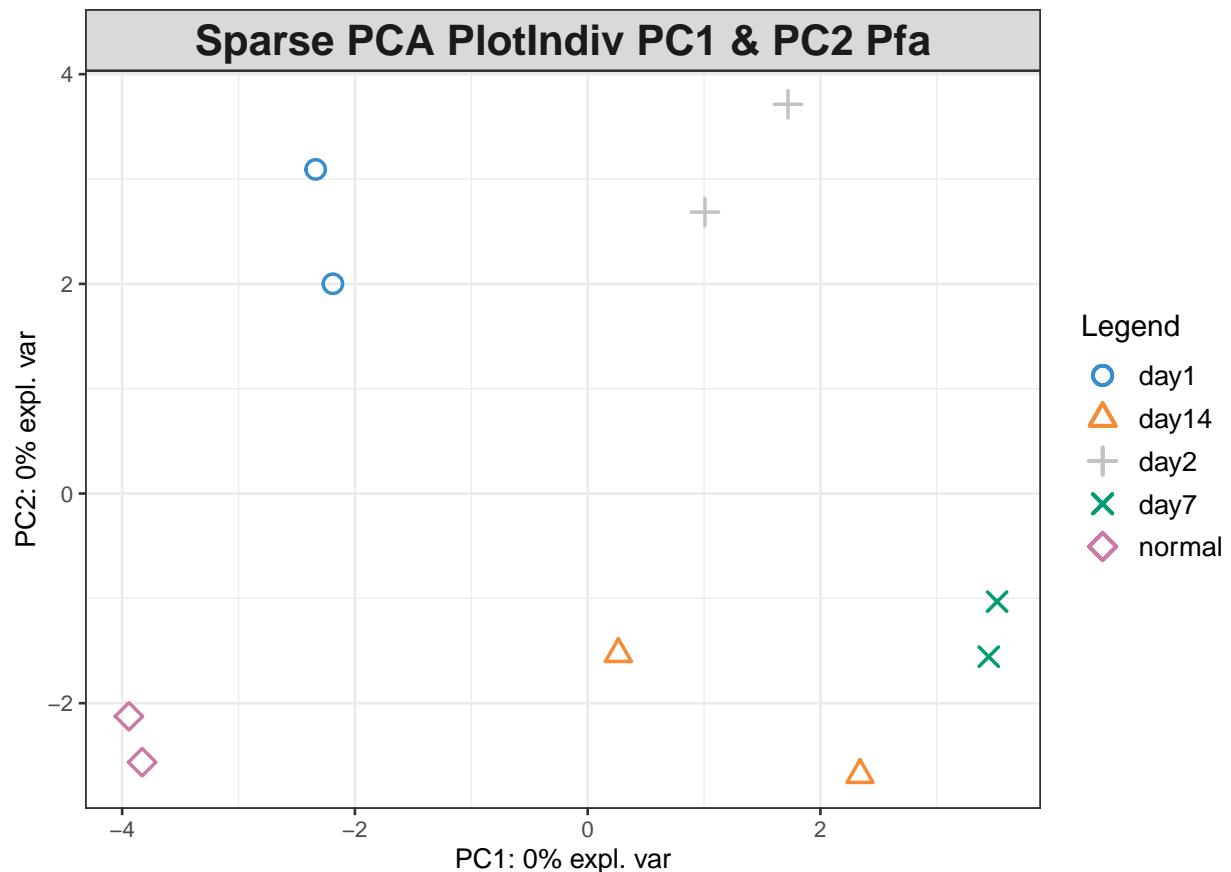
spca.fa_t <- spca(fa_t, ncomp=3, keepX=c(10,10,10))
spca.pfa_t <- spca(pfa_t, ncomp=3, keepX=c(10,10,10))
#?spca

plotIndiv(spca.fa_t, group = metadata$condition, comp= c(1,2), ind.names = FALSE, legend=TRUE, title = "PCA PlotIndiv PC1 & PC2 pfa")

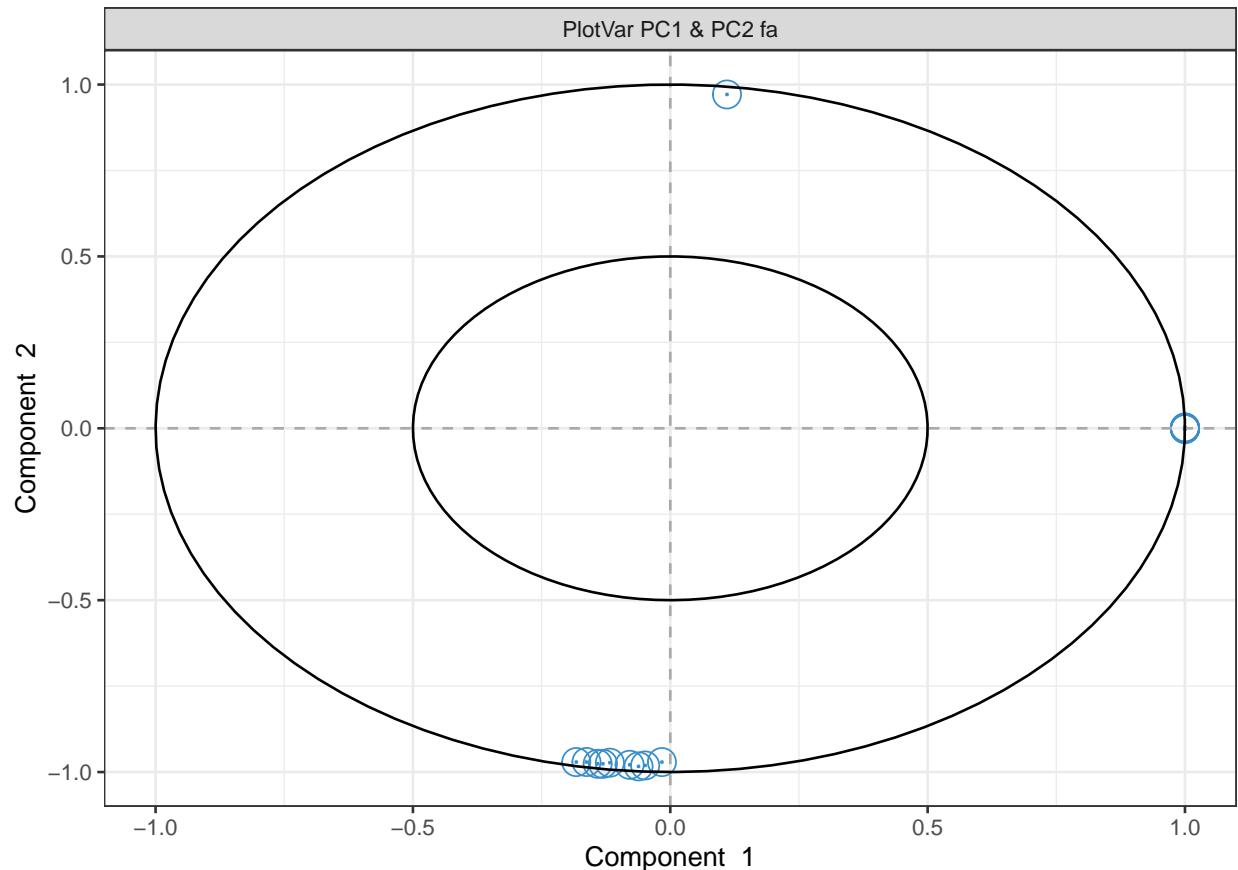
```



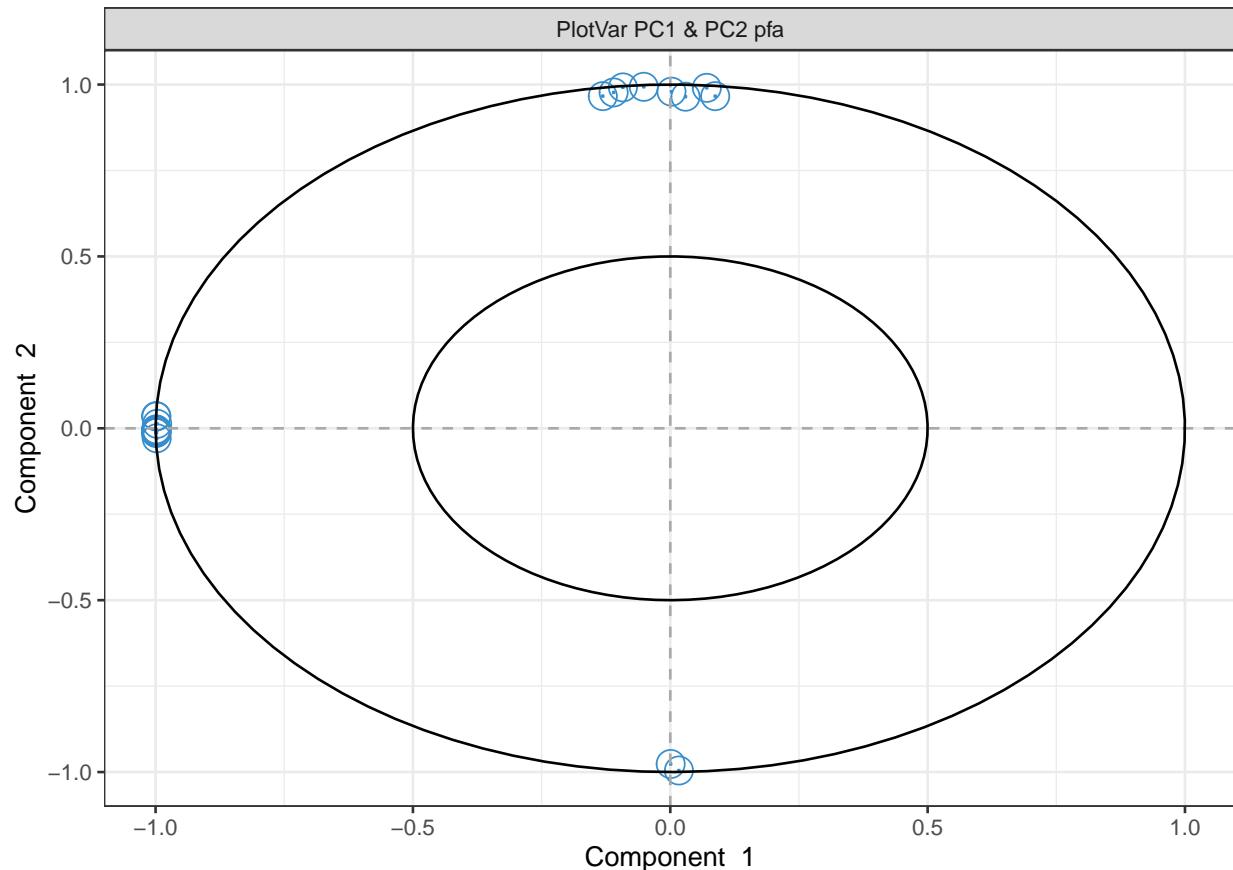
```
plotIndiv(sPCA.pfa_t, group = metadata$condition, comp= c(1,2), ind.names = FALSE, legend=TRUE, title = "Sparse PCA PlotIndiv PC1 & PC2 fa")
```



```
plotVar(sPCA.fa_t, var.names = FALSE, comp= c(1,2), title = "PlotVar PC1 & PC2 fa")
```



```
plotVar(sPCA.pfa_t, var.names = FALSE, comp= c(1,2), title = "PlotVar PC1 & PC2 pfa")
```



## PLS-DA

### Analyses multivariées supervisées

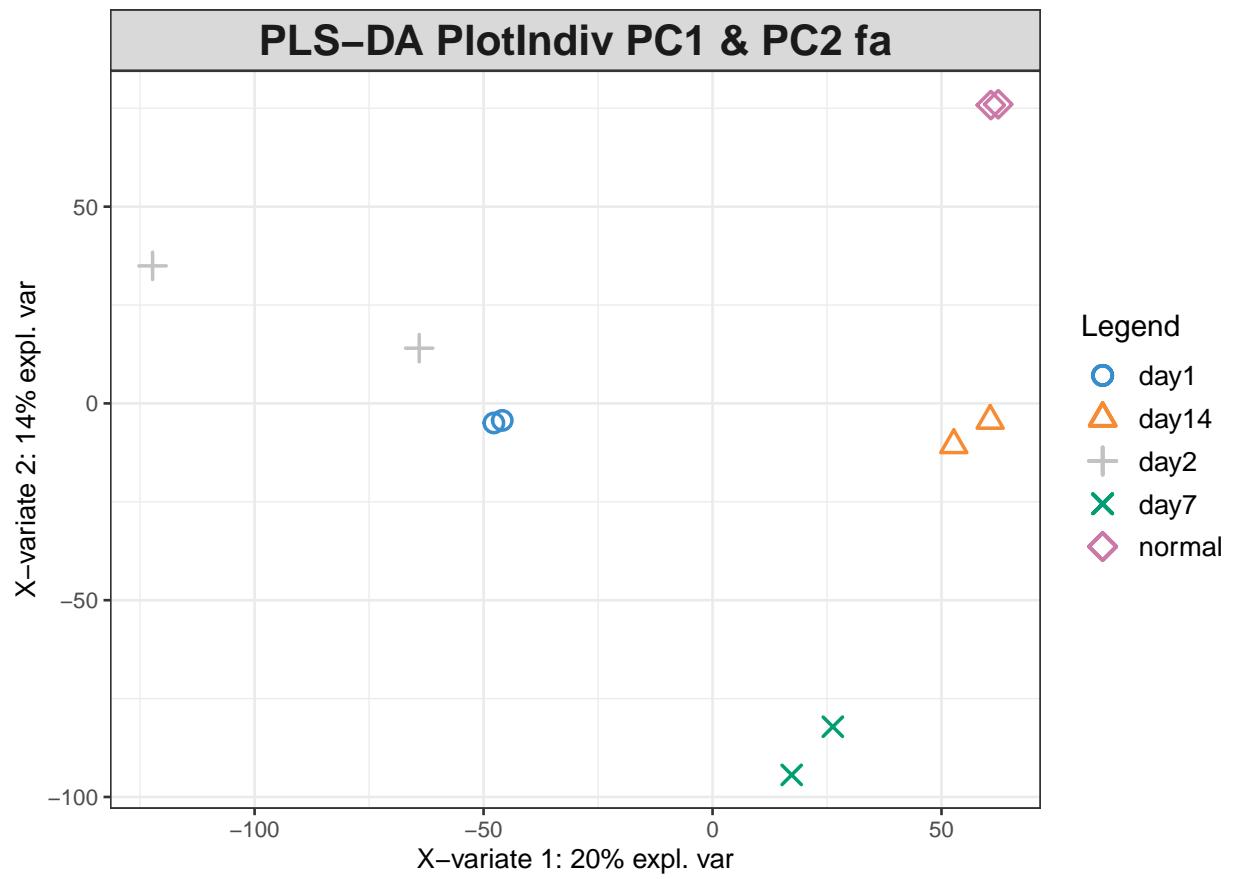
```

W <- metadata$condition

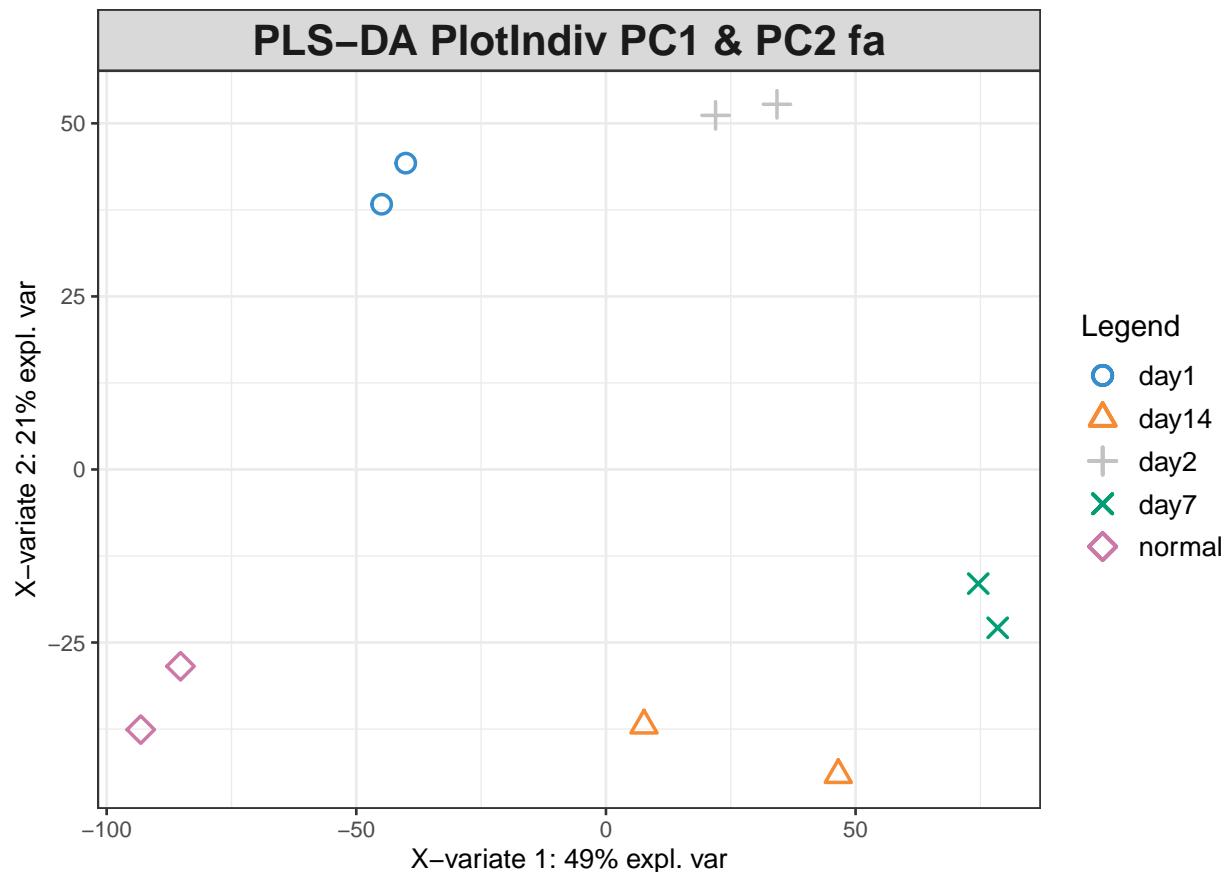
plsda.fa_t <- mixOmics::plsda(fa_t, W, ncomp = 3)
plsda.pfa_t <- mixOmics::plsda(pfa_t, W, ncomp = 3)

#Error: could not find function "plsda" donc rajouter mixOmics::
mixOmics::plotIndiv(plsda.fa_t, comp= c(1,2), ind.names=FALSE, legend=TRUE, title = "PLS-DA PlotIndiv PC")

```

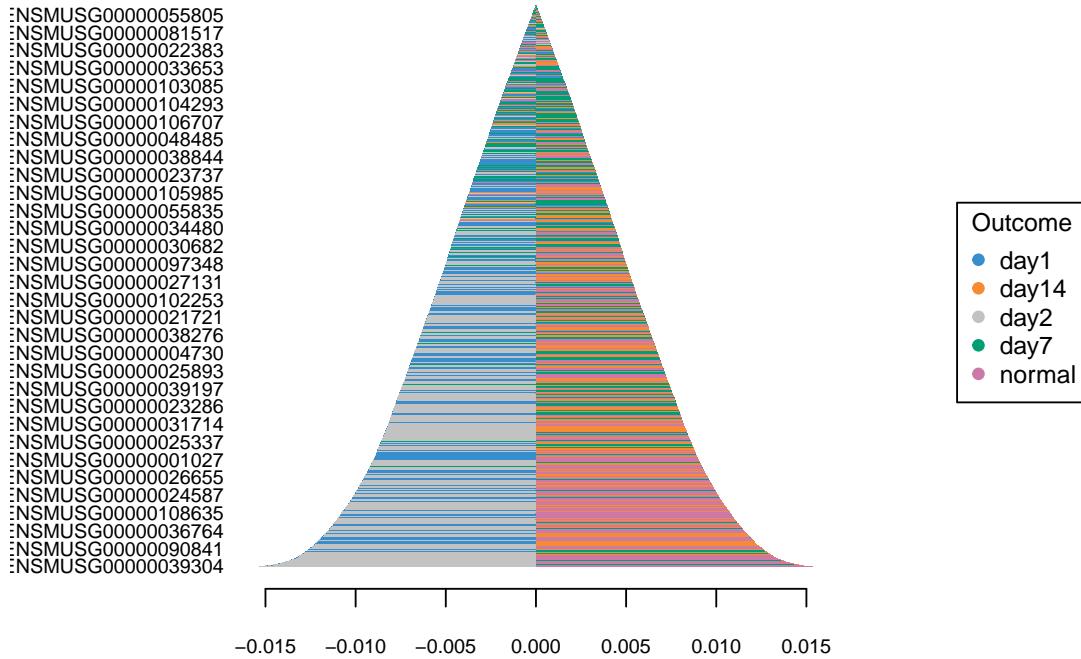


```
mixOmics::plotIndiv(plsda.pfa_t, comp= c(1,2), ind.names=FALSE, legend=TRUE, title = "PLS-DA PlotIndiv PC1 & PC2 fa")
```



```
mixOmics::plotLoadings(plsda.fa_t, comp=1, contrib = "max")
```

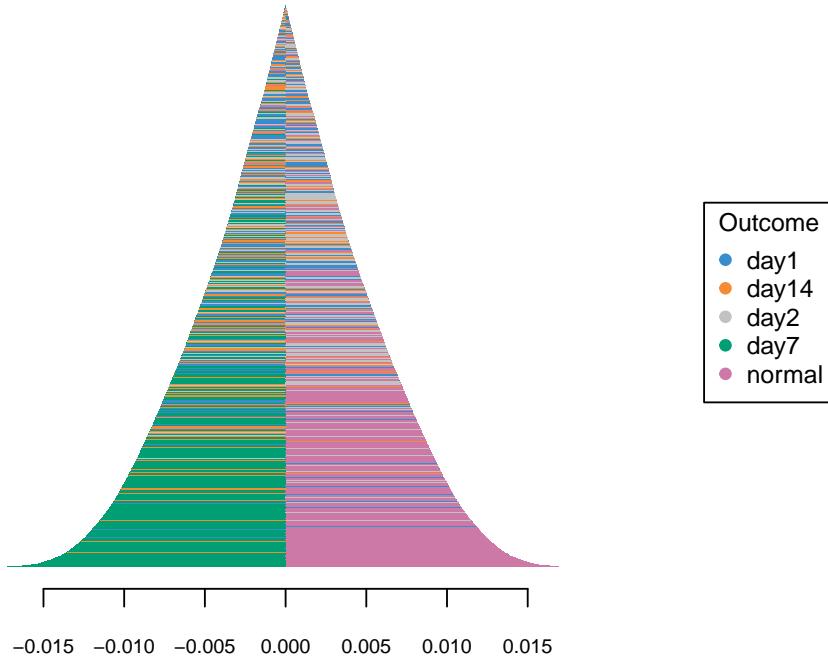
## Contribution on comp 1



```
mixOmics::plotLoadings(plsda.fa_t, comp=2, contrib = "max")
```

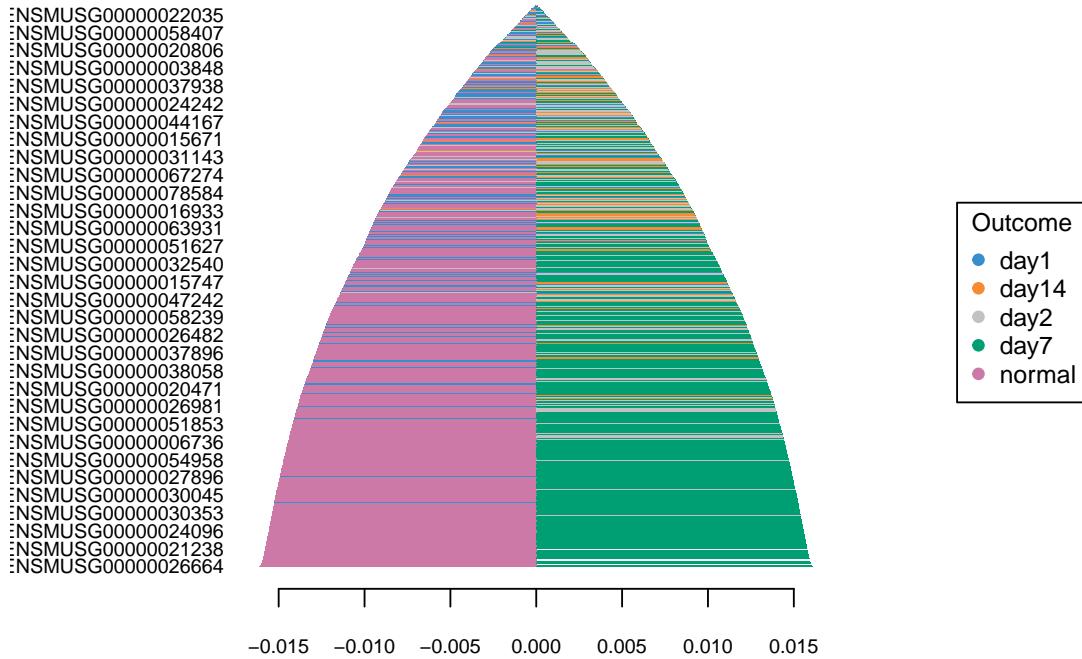
## Contribution on comp 2

```
:NSMUSG00000103864  
:NSMUSG0000091780  
:NSMUSG0000087968  
:NSMUSG0000035798  
:NSMUSG0000040601  
:NSMUSG0000054836  
:NSMUSG0000028318  
:NSMUSG0000107643  
:NSMUSG0000042302  
:NSMUSG0000086487  
:NSMUSG0000092275  
:NSMUSG0000043760  
:NSMUSG0000031860  
:NSMUSG0000022359  
:NSMUSG0000092592  
:NSMUSG0000031390  
:NSMUSG0000093780  
:NSMUSG0000027502  
:NSMUSG00000102139  
:NSMUSG0000004668  
:NSMUSG0000046456  
:NSMUSG0000022864  
:NSMUSG0000064330  
:NSMUSG0000037035  
:NSMUSG0000096975  
:NSMUSG0000022723  
:NSMUSG0000024181  
:NSMUSG0000030309  
:NSMUSG00000104086  
:NSMUSG0000048000  
:NSMUSG0000021484  
:NSMUSG0000037649
```



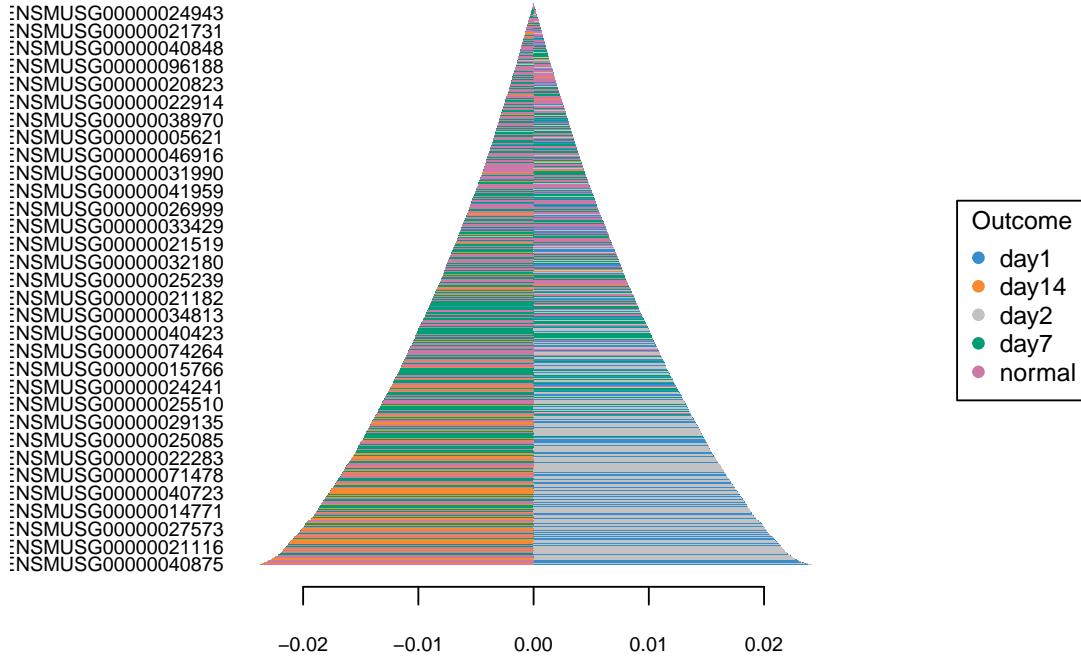
```
mixOmics::plotLoadings(plsda.pfa_t, comp=1, contrib = "max")
```

## Contribution on comp 1



```
mixOmics::plotLoadings(plsda.pfa_t, comp=2, contrib = "max")
```

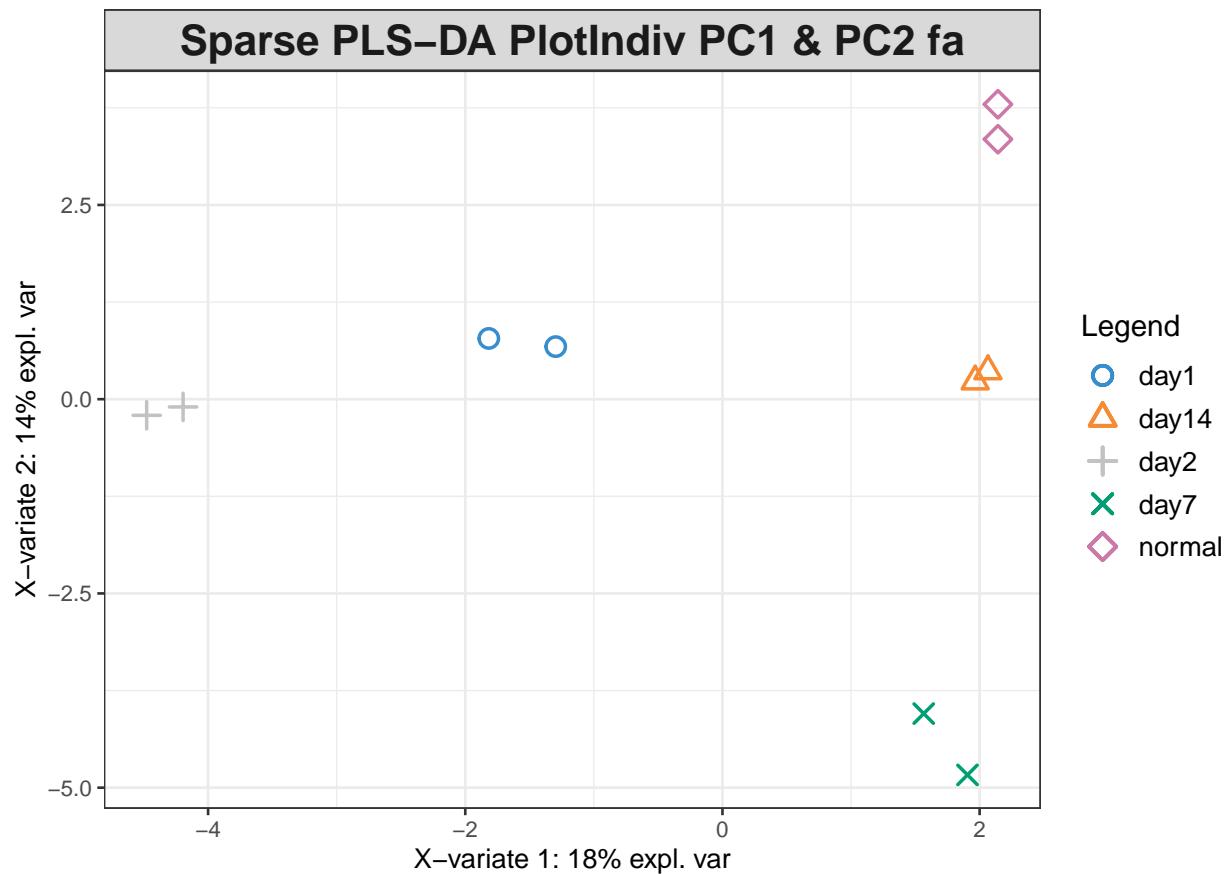
## Contribution on comp 2



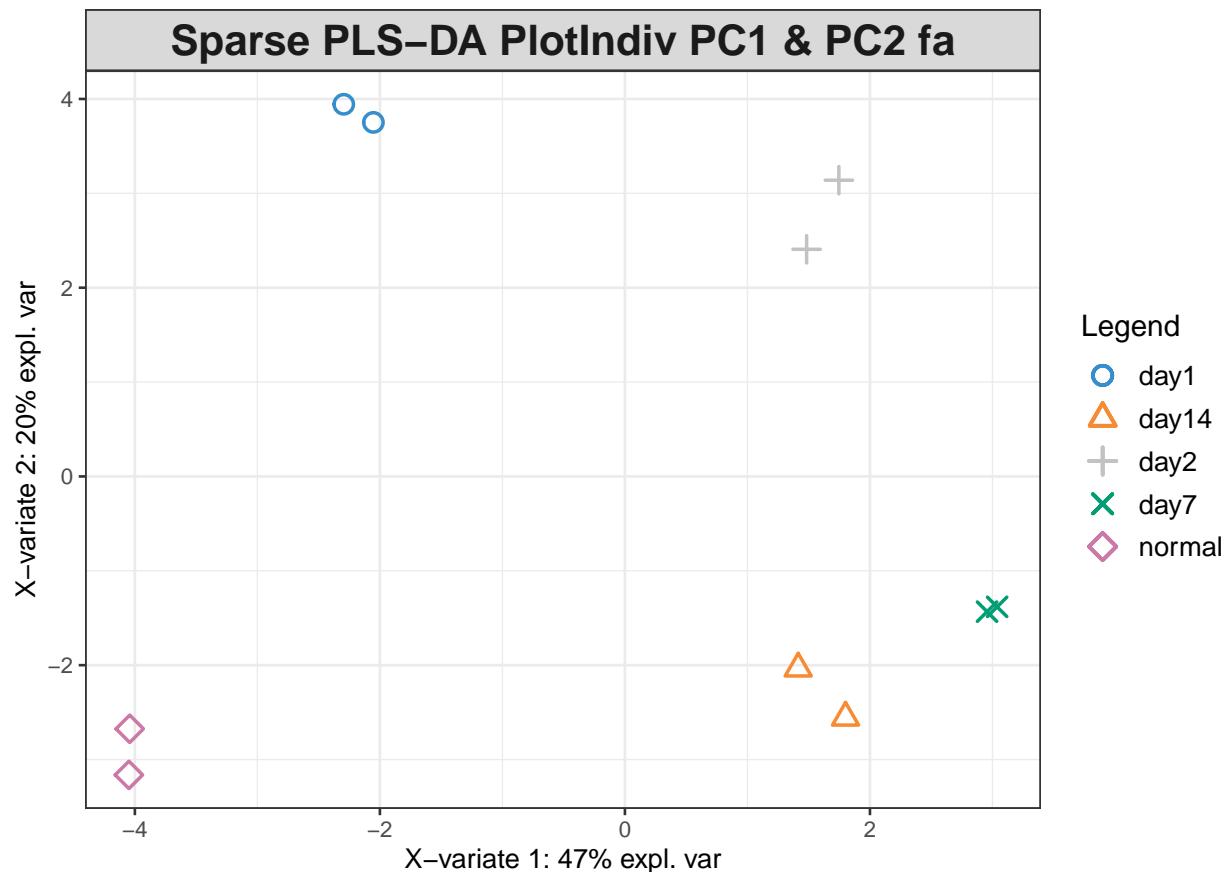
## Sparse PLS-DA

```
splsda.fa_t <- splsda(fa_t, W, ncomp = 3, keepX = c(10,10,10))  
splsda.pfa_t <- splsda(pfa_t, W, ncomp = 3, keepX = c(10,10,10))
```

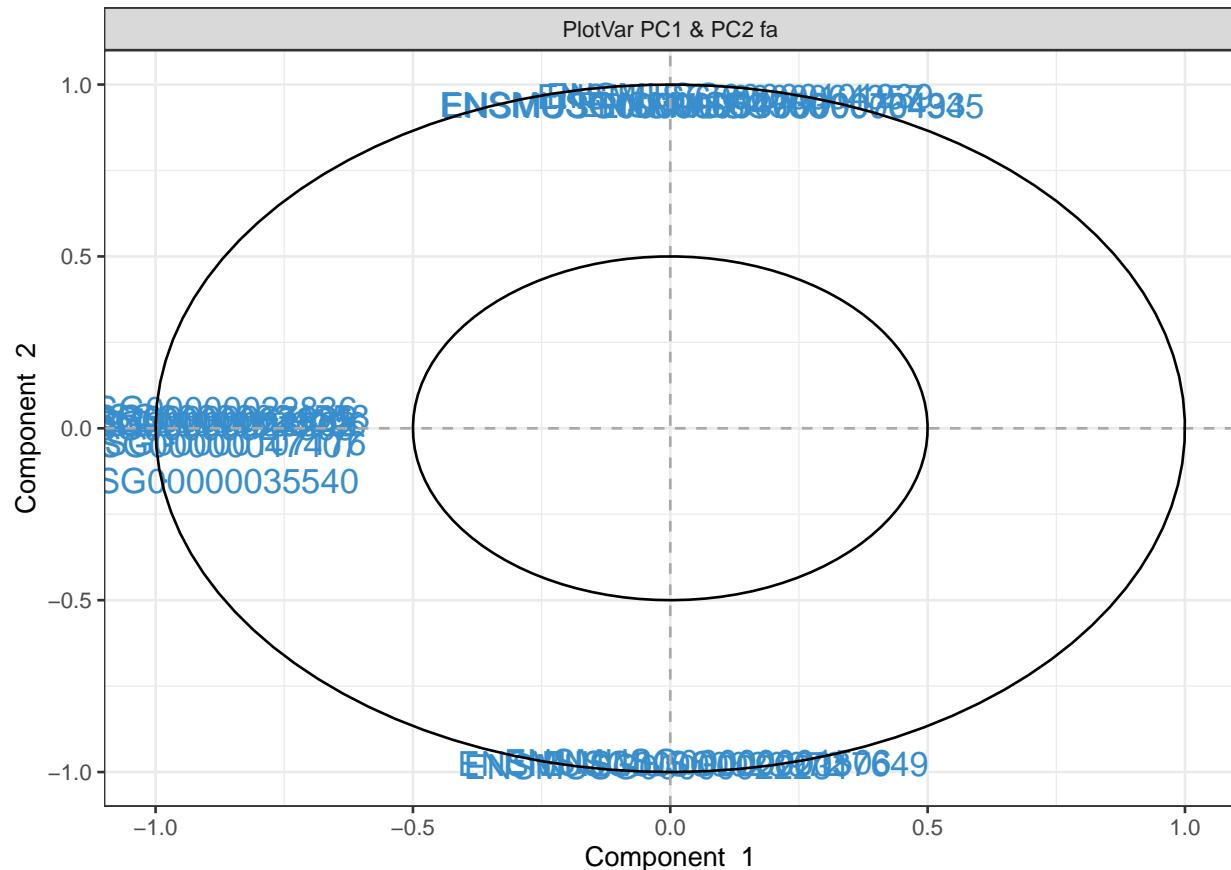
```
plotIndiv(splsda.fa_t, ind.names=FALSE, legend=TRUE, comp = c(1,2), title = "Sparse PLS-DA PlotIndiv PC1 & PC2")
```



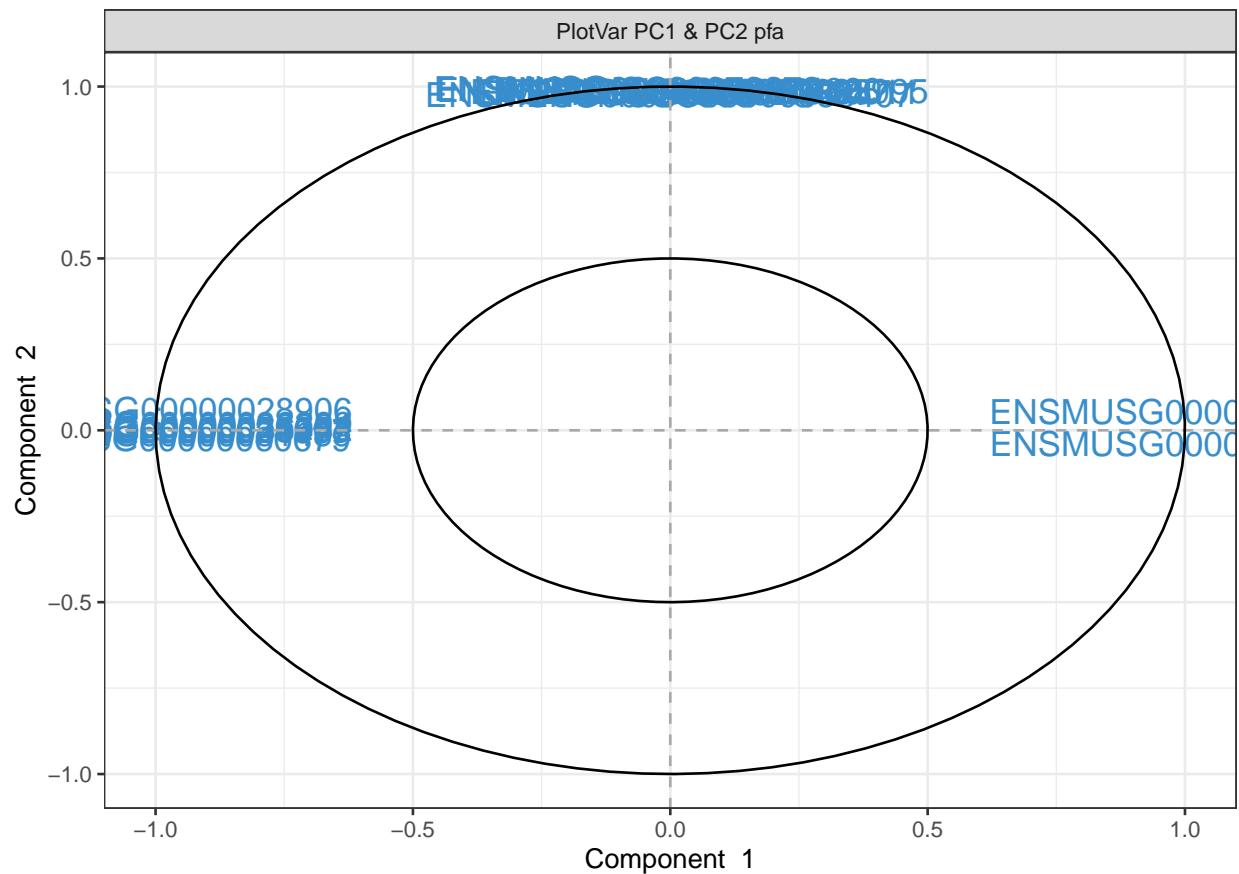
```
plotIndiv(splsda.pfa_t, ind.names=FALSE, legend=TRUE, comp = c(1,2), title = "Sparse PLS-DA PlotIndiv PC1 & PC2 fa")
```



```
plotVar(splsda.fa_t, var.names = TRUE, comp= c(1,2),title = "PlotVar PC1 & PC2 fa")
```

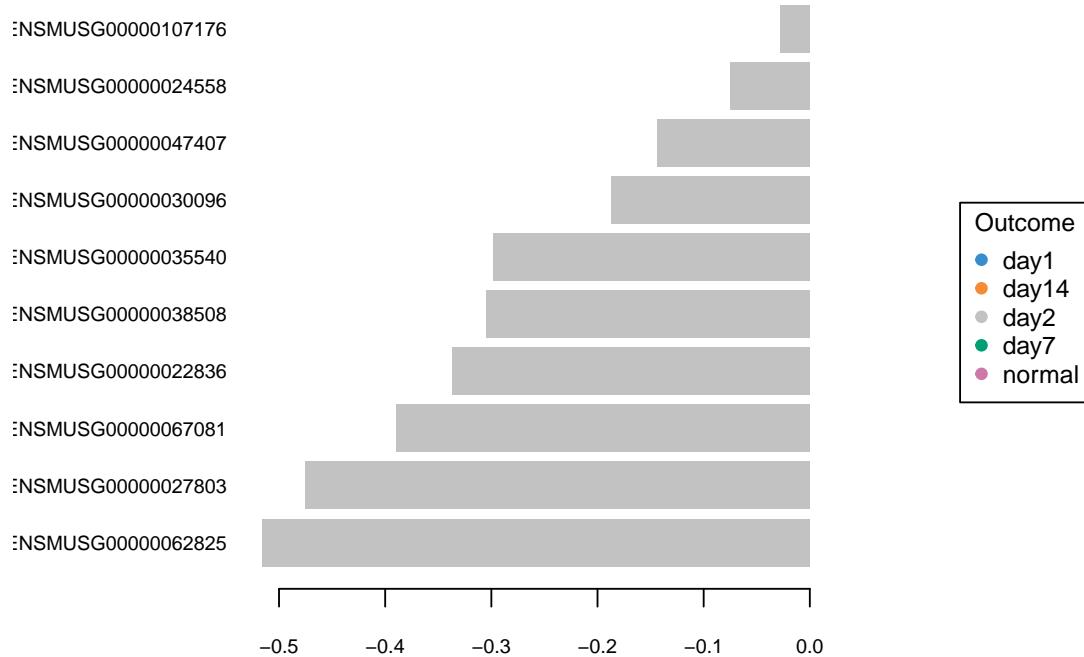


```
plotVar(splsda.pfa_t, var.names = TRUE, comp= c(1,2),title = "PlotVar PC1 & PC2 pfa")
```



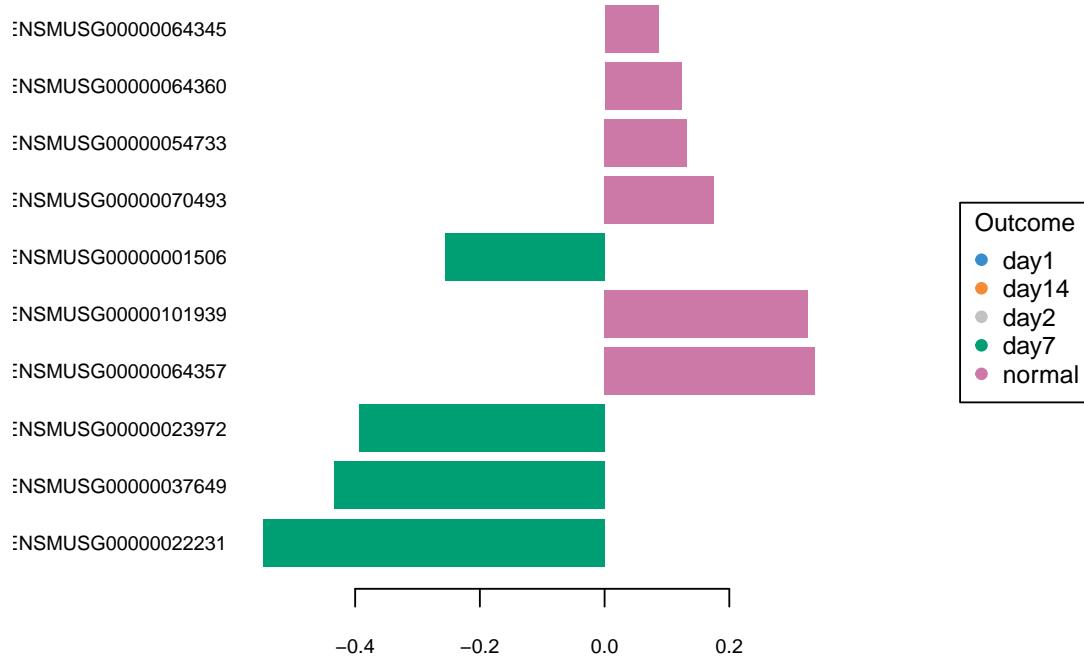
```
plotLoadings(splsda.fa_t, comp=1, contrib = 'max')
```

## Contribution on comp 1



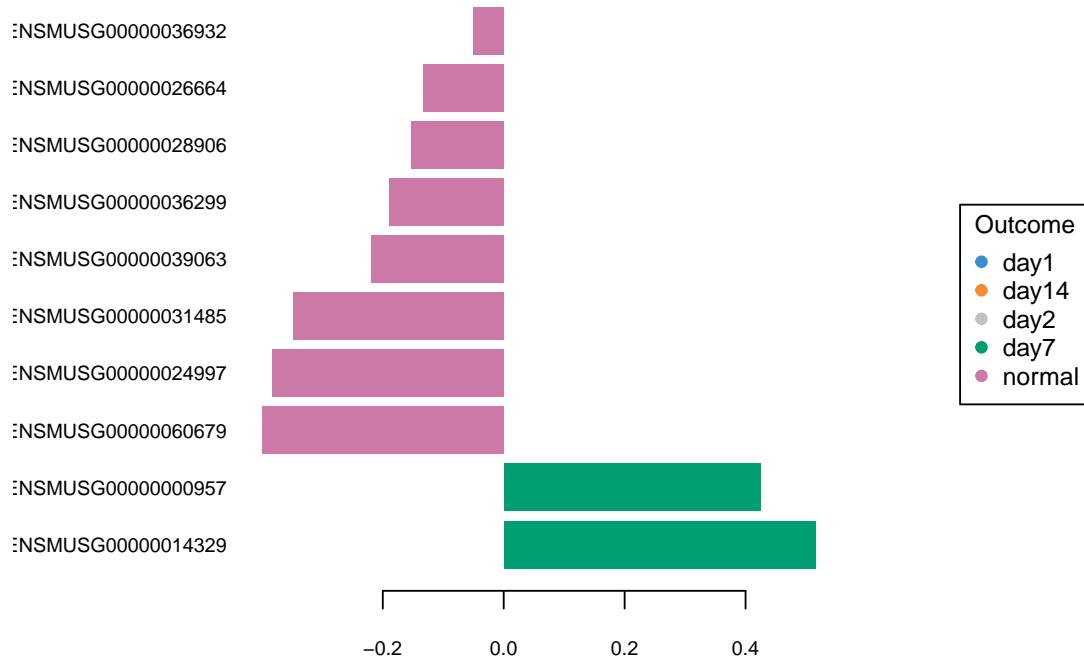
```
plotLoadings(splsda.fa_t, comp=2, contrib = 'max')
```

## Contribution on comp 2



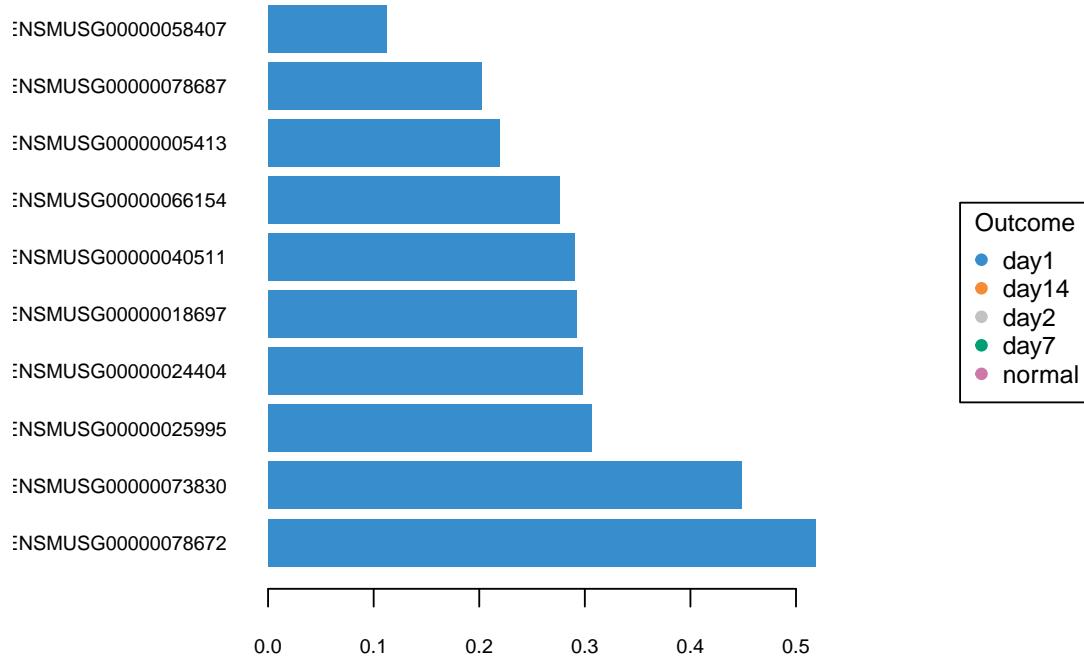
```
plotLoadings(splsda.pfa_t, comp=1, contrib = 'max')
```

## Contribution on comp 1



```
plotLoadings(splsda.pfa_t, comp=2, contrib = 'max')
```

## Contribution on comp 2

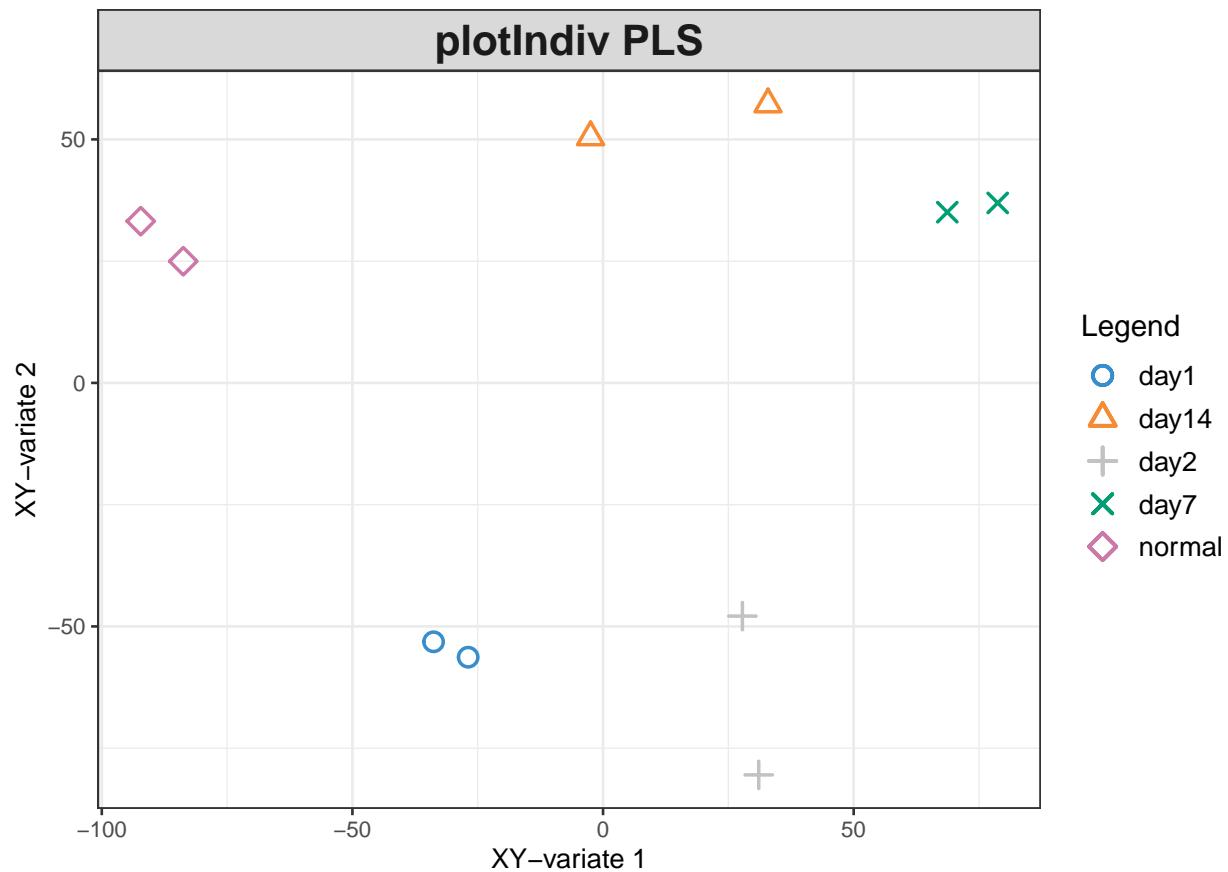


## Analyses multi omics

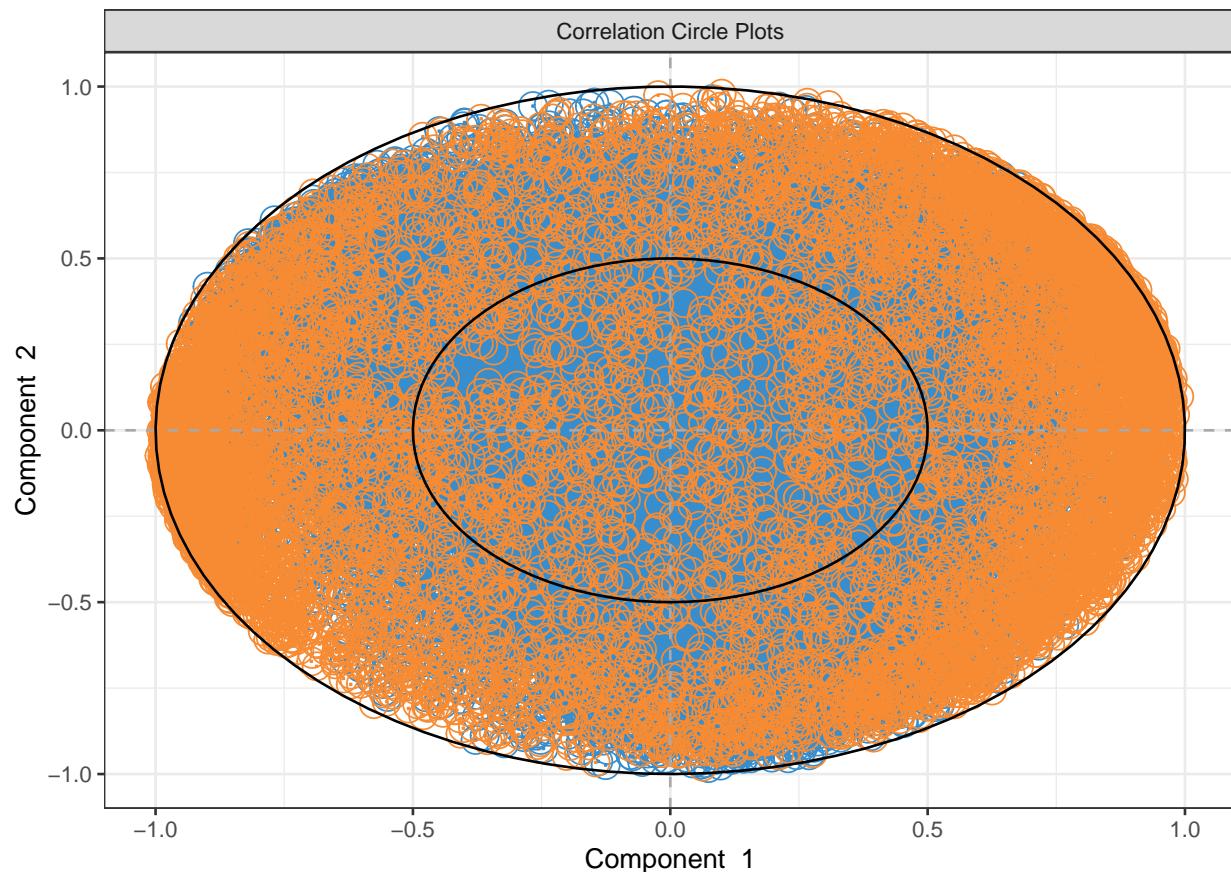
### PLS

```
pls.fa.pfa <- pls(fa_t,pfa_t)

plotIndiv(pls.fa.pfa,rep.space="XY-variate",
          title = "plotIndiv PLS",
          ind.names=FALSE,
          group=metadata$condition,
          pch = as.numeric(factor(metadata$condition)),
          pch.levels =metadata$condition,
          legend = TRUE)
```



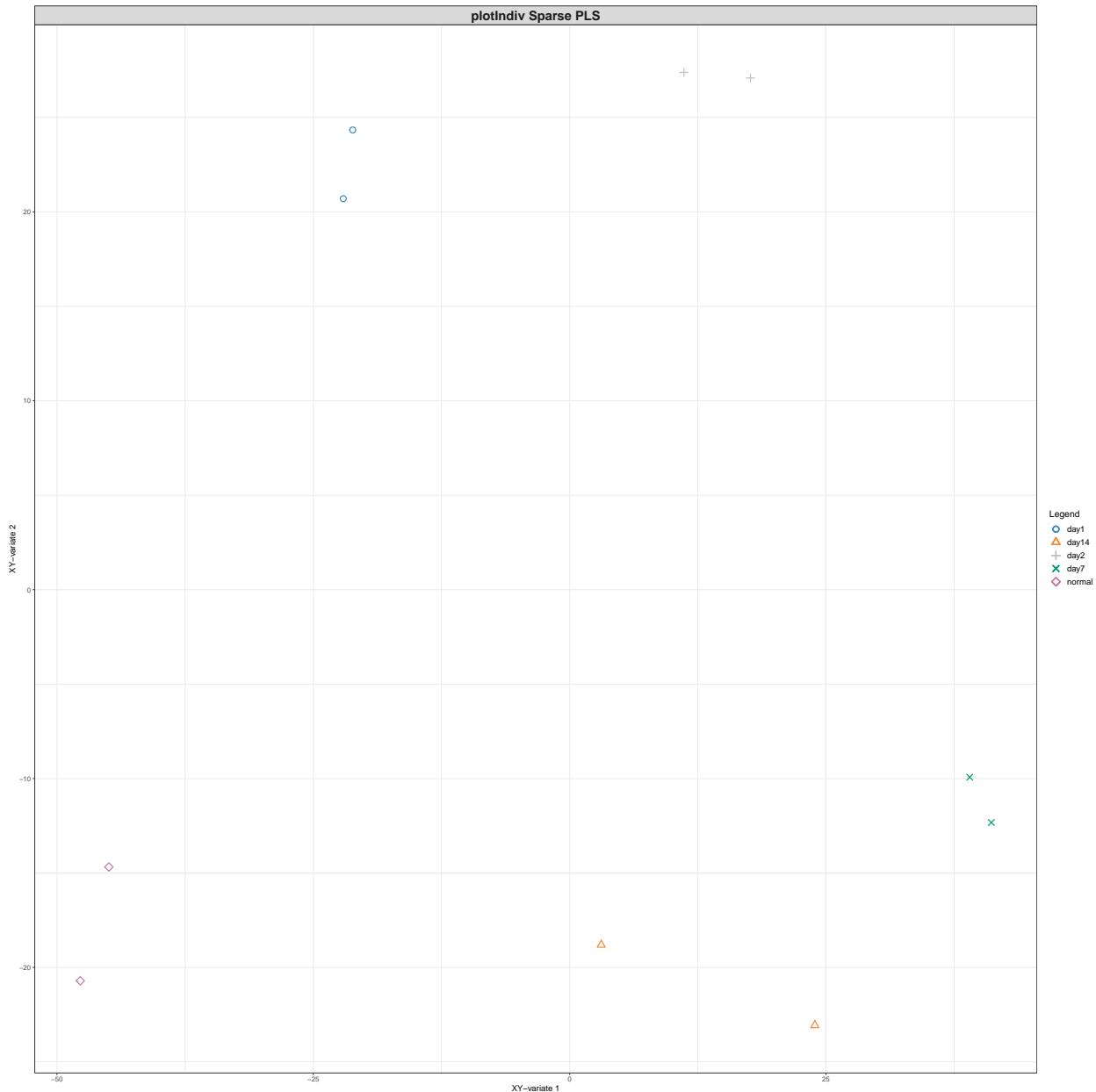
```
plotVar(pls.fa.pfa, var.names = c(FALSE, FALSE))
```



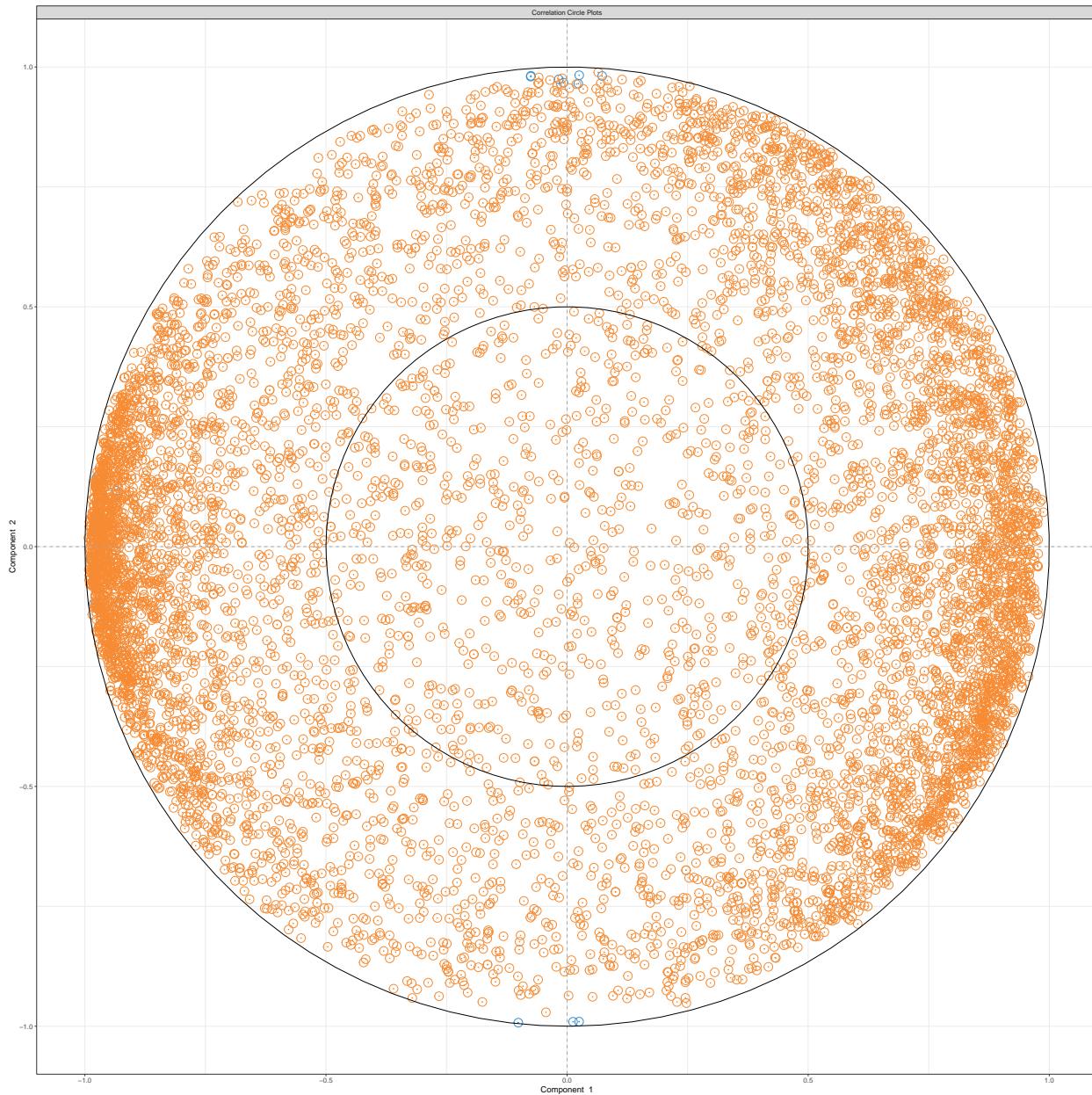
## S-PLS

```
spls.fa.pfa <- spls(fa_t,pfa_t, ncomp=10, keepX = c(10,10,10))

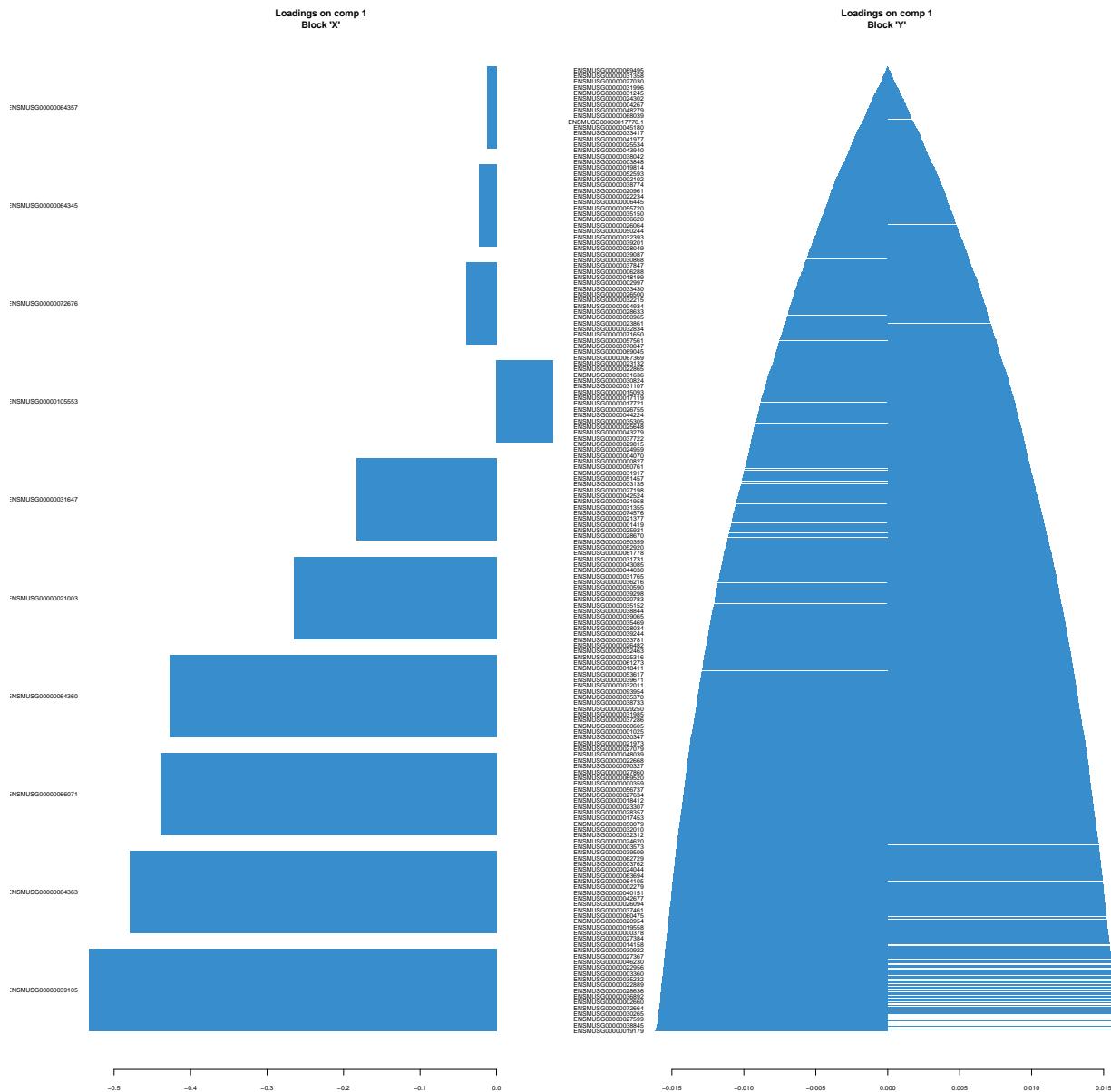
plotIndiv(spls.fa.pfa, rep.space="XY-variate",
          title = "plotIndiv Sparse PLS",
          ind.names=FALSE,
          group=metadata$condition,
          pch = as.numeric(factor(metadata$condition)),
          pch.levels =metadata$condition,
          legend = TRUE)
```



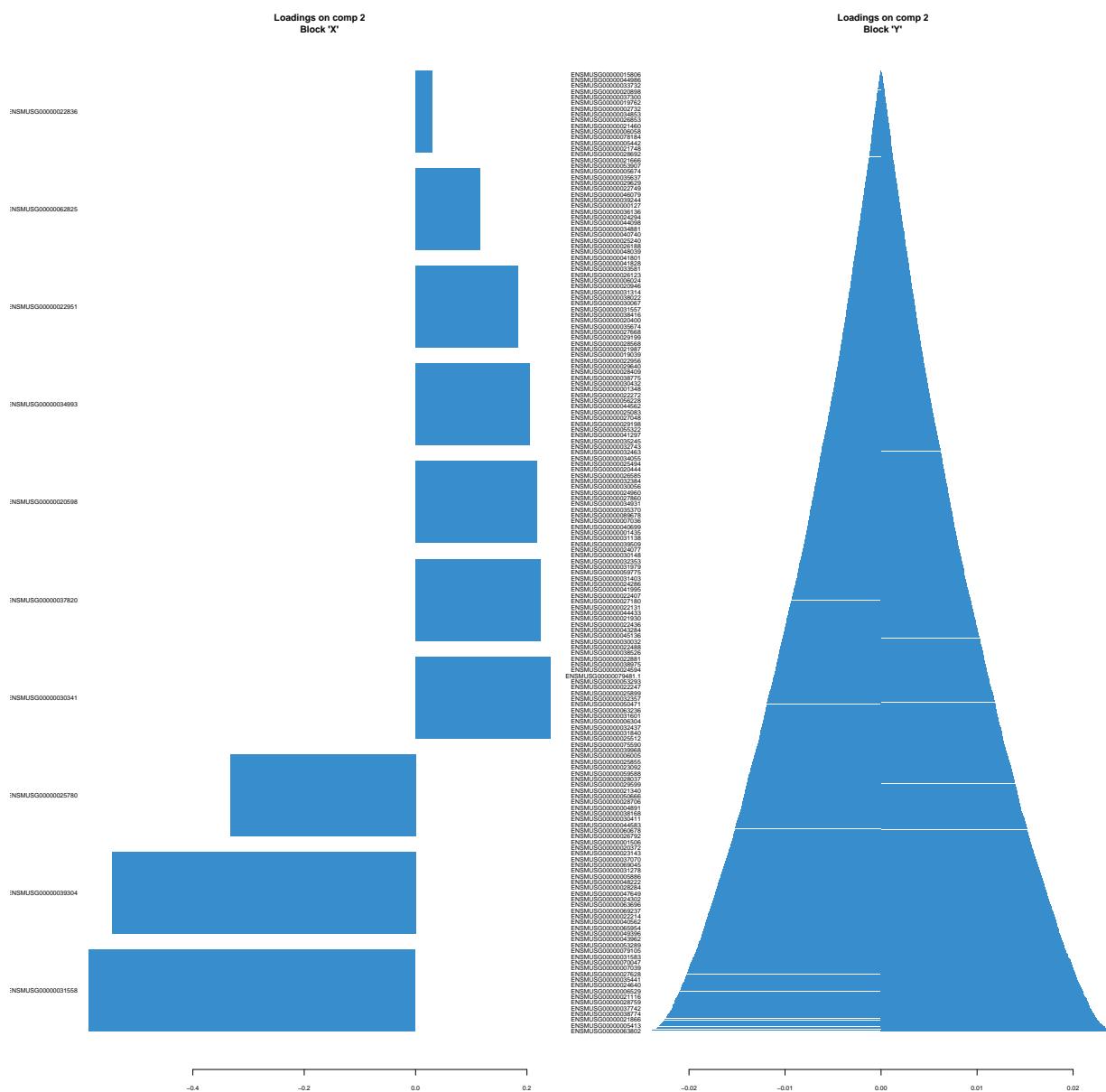
```
plotVar(spls.fa.pfa, var.names = c(FALSE, FALSE))
```



```
plotLoadings(spls.fa.pfa, comp=1, size.title = 1, name.var = NULL, max.name.length = 50)
```



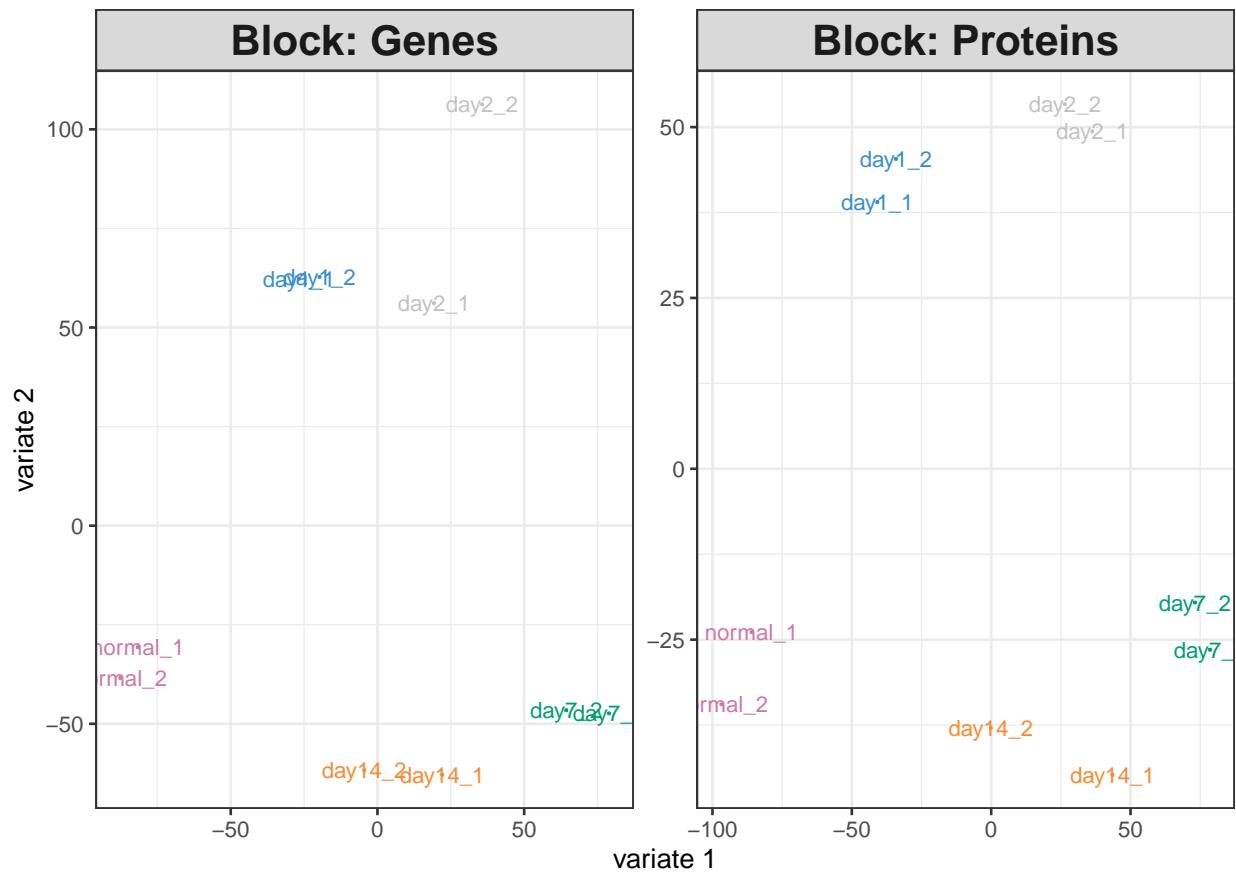
```
plotLoadings(spls.fa.pfa, comp=2, size.title = 1, name.var = NULL, max.name.length = 50)
```



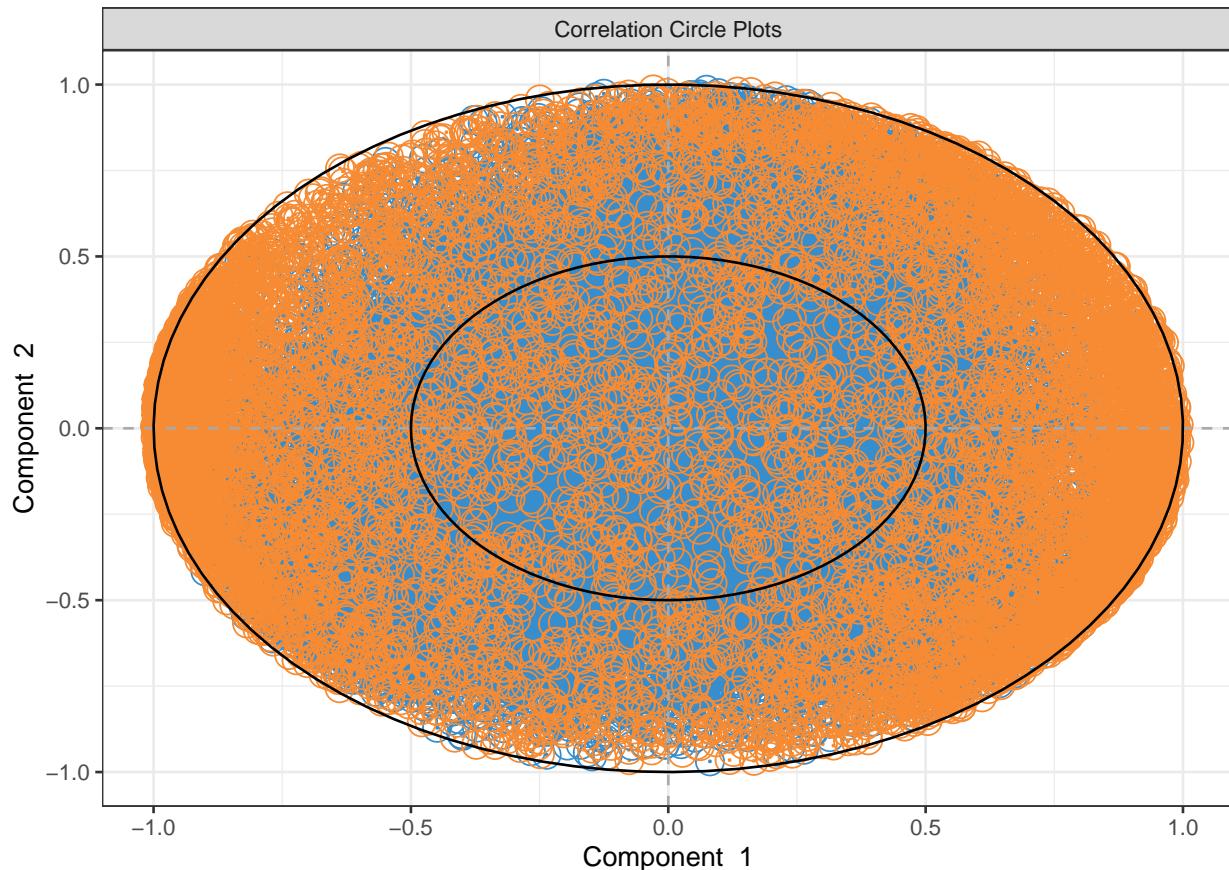
## Multi-block PLS-DA

```
block.plsda.fa.pfa <- block.plsda(
  X = list(Genes = fa_t,
           Proteins = pfa_t),
  Y = metadata$condition)

plotIndiv(block.plsda.fa.pfa)
```



```
plotVar(block.plsda.fa.pfa, var.names = c(FALSE, FALSE))
```



### Multi-block S-PLS-DA / DIABLO

```
data <- list(Genes = fa_t,
             Proteins = pfa_t)

lapply(data, dim)
```

```
$Genes
[1] 10 22459
```

```
$Proteins
[1] 10 8044
```

```
Y <- metadata$condition
Y
```

```
[1] "normal" "normal" "day1"    "day1"    "day2"    "day2"    "day7"    "day7"    "day14"   "day14"
```

```
design = matrix(0.1, ncol = length(data), nrow = length(data),
               dimnames = list(names(data), names(data)))
```

```
diag(design) = 0
design
```

## Design

	Genes	Proteins
Genes	0.0	0.1
Proteins	0.1	0.0

Tout d'abord, nous ajustons un modèle DIABLO sans sélection de variable pour évaluer la performance globale et choisir le nombre de composants pour le modèle DIABLO final. La fonction perf est exécutée avec une validation croisée 10 fois répétée 10 fois.

```
sgccda.res = block.splsda(X = data, Y = Y, ncomp = 10, design = design)

#set.seed(123) # for reproducibility, only when the `cpus` argument is not used
# this code takes a couple of min to run
#perf.diablo = perf(sgccda.res, validation = 'Mfold', folds = 2, nrepeat = 3)

#perf.diablo # lists the different outputs
#plot(perf.diablo)
```

```
#perf.diablo$choice.ncomp$WeightedVote
```

```
#ncomp = perf.diablo$choice.ncomp$WeightedVote["Overall.BER", "centroids.dist"]
#ncomp
```

Le nombre de composantes à garder est de 4

**Tuning keepX** Je choisis le nombre optimal de variables à sélectionner dans chaque ensemble de données à l'aide de la fonction tune.block.splsda.

```
# #set.seed(123) # for reproducibility, only when the `cpus` argument is not used
# test.keepX = list (microbiote = c(1:9, seq(10, 45, 5), seq(50,150,10)),
#                     caecum = c(1:9, seq(10, 45, 5), seq(50,150,10)),
#                     hypothalamus = c(1:9, seq(10,45 , 5), seq(50,150,10)))
#
# tune.TCGA = tune.block.splsda(X = data, Y = Y, ncomp = ncomp,
#                               test.keepX = test.keepX, design = design,
#                               validation = 'Mfold', folds = 10, nrepeat = 1,
#                               dist = "centroids.dist")
#
#
# list.keepX = tune.TCGA$choice.keepX
#
# #qsub -cwd -V -N test_keepX -q long.q -pe thread 4 -o test_keepX.out -e test_keepX.err -b y "Rscript -
#
# list.keepX <- list(Genes = c(4, 6,5,5), Proteins = c(5,7,5,6))
list.keepX
```

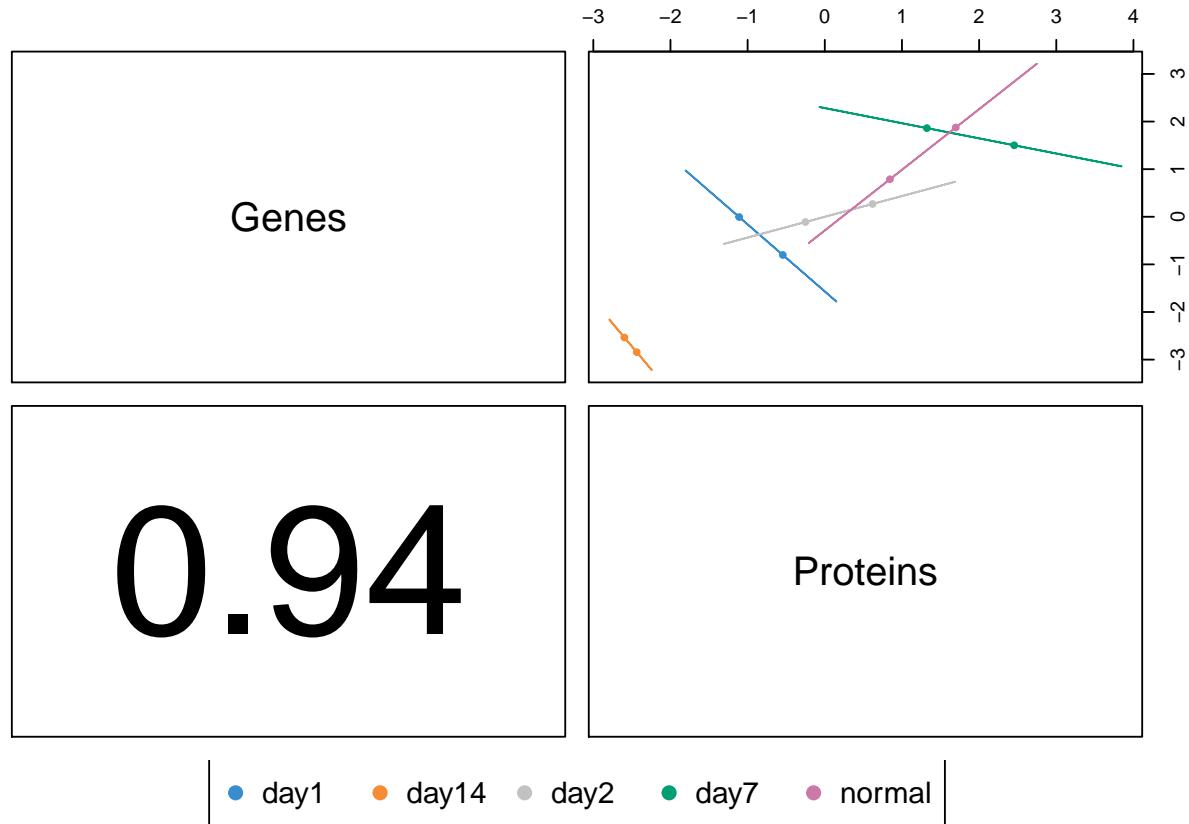
```
$Genes  
[1] 4 6 5 5
```

```
$Proteins  
[1] 5 7 5 6
```

```
sgccda.res = block.splsda(X = data, Y = Y, ncomp = 4,  
                           keepX = list.keepX, design = design)
```

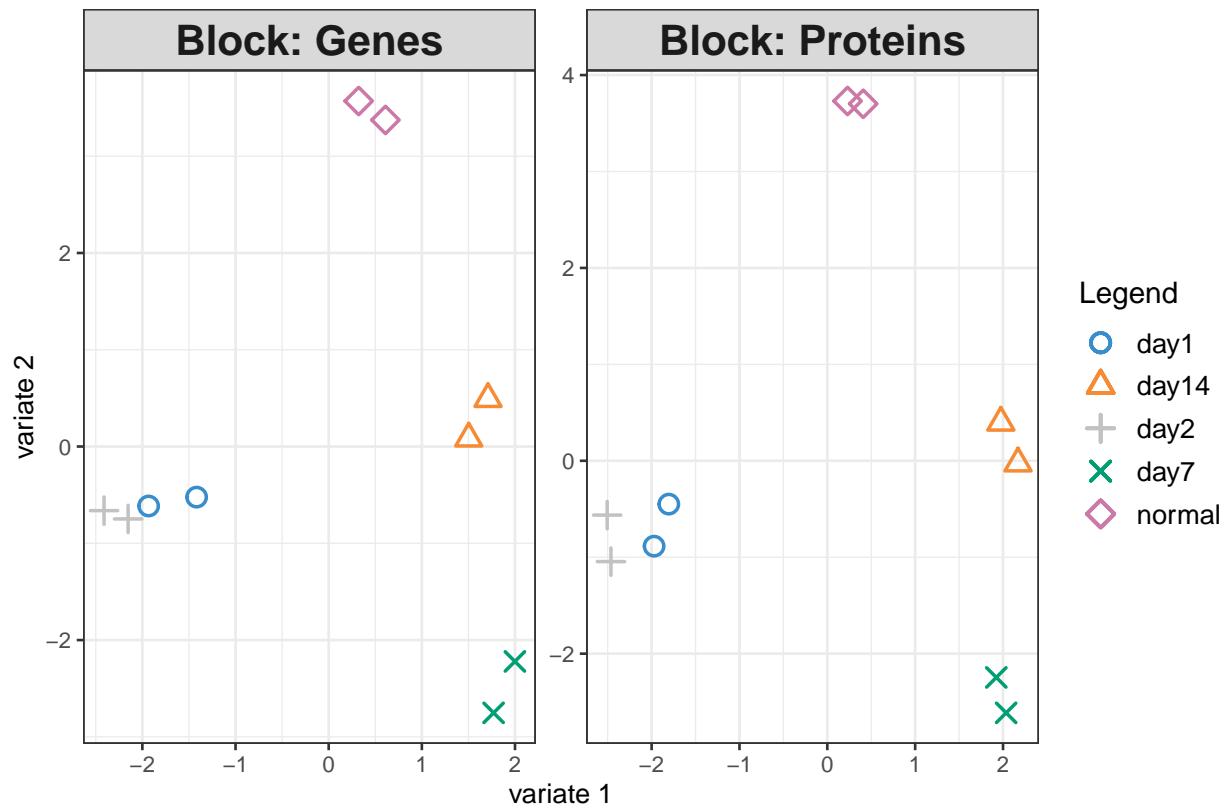
**Final model** plotDIABLO est un tracé de diagnostic pour vérifier si la corrélation entre les composants de chaque ensemble de données a été maximisée comme spécifié dans la matrice de conception.

```
plotDiabIo(sgccda.res, ncomp = 4)
```



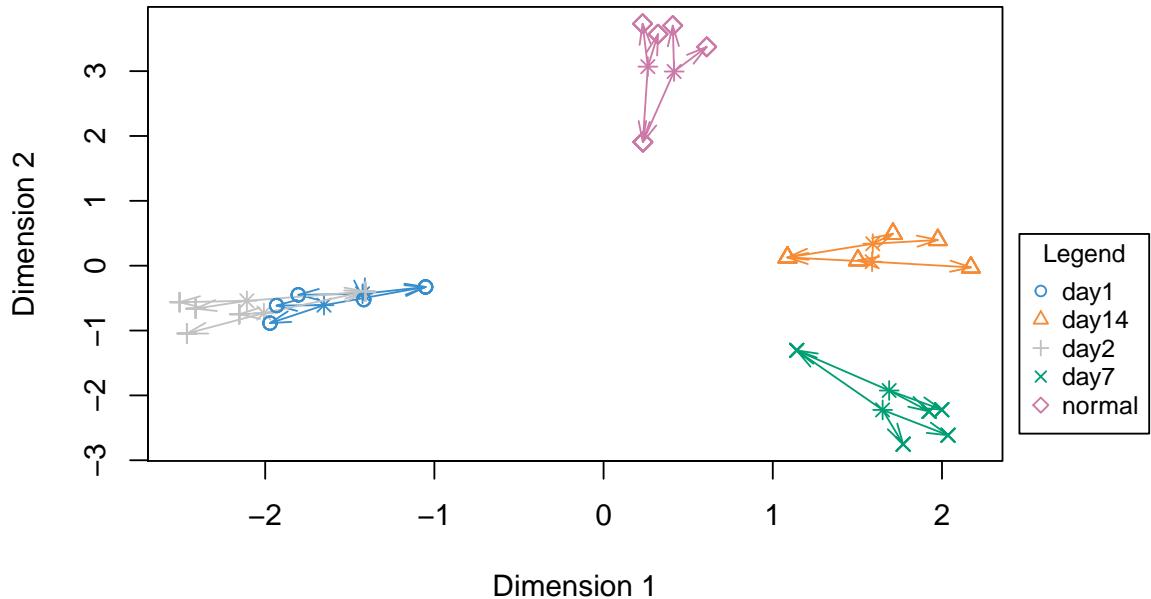
```
plotIndiv(sgccda.res, ind.names = FALSE, legend = TRUE, title = 'DIABLO')
```

# DIABLO



```
plotArrow(sgccda.res, ind.names = FALSE, legend = TRUE, title = 'DIABLO')
```

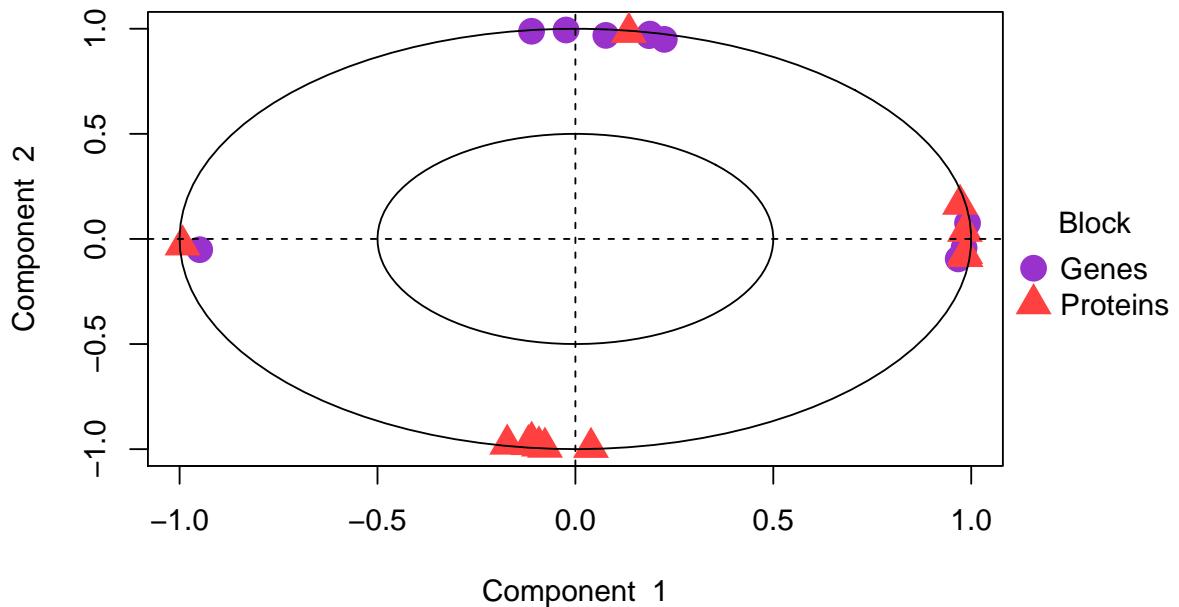
## DIABLO



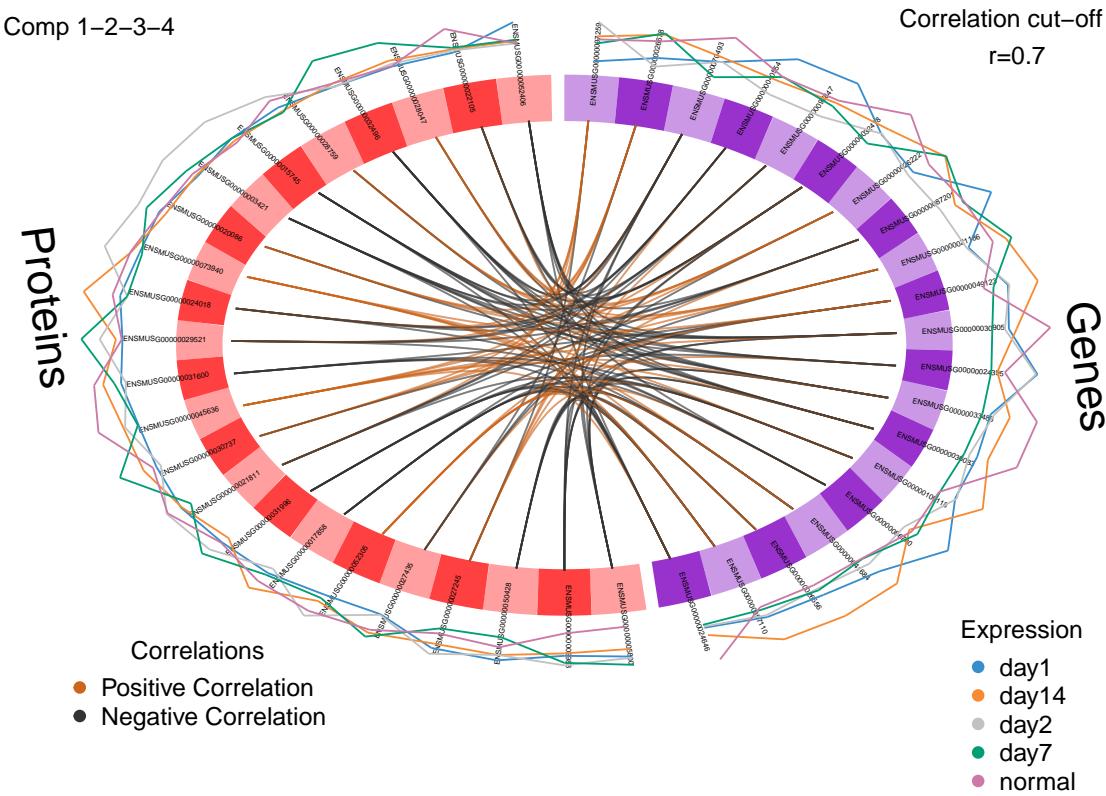
```
plotVar(sgccda.res, var.names = FALSE, style = 'graphics', legend = TRUE,
        pch = c(16, 17), cex = c(2,2), col = c('darkorchid', 'brown1'))
```

Variable plots

## Correlation Circle Plots



```
circosPlot(sgccda.res, cutoff = 0.7, line = TRUE,
           color.blocks= c('darkorchid', 'brown1'),
           color.cor = c("chocolate3","grey20"), size.labels = 1.5)
```



## WGCNA

Reconstruct the co-expression network from all the time points of the FA transcriptomics data. Propose to filter and remove all the zero expressed genes, the NAs and the less informative genes from the transcriptomics data. (I remove all the genes that are not expressed in at least 9 out of the 18 conditions (expression > 1 TPM in 9) and then filter with the coefficient of variation > 0.75).

Then apply the first part of the network reconstruction steps as we saw them on the WGCNA course until the module predictions.

Instead of using WGCNA's module prediction routines, apply a universal threshold of 0.5 on the adjacency matrix, and obtain an adjacency matrix that is reduced in size. This is the network.

```
#Load the WGCNA package
library(WGCNA)
```

J'utilise les données de transcriptomique **fa** **normalisées**

J'harmonise les row.names entre la count table et les metadata

```
head(fa_expr_norm)
```

	normal_1	normal_2	normal_3	day1_1	day1_2	day1_3	day2_1
ENSMUSG000000000001	719.032852	1247.299971	809.375515	2711.22417	1094.745589	1331.5880610	1185.658698

```

ENSMUSG000000000003  0.000000  0.000000  0.000000  0.000000  0.000000  0.0000000  0.000000
ENSMUSG000000000028  10.420766  7.448588  1.700369  42.82759   13.485108  21.9214582  166.379146
ENSMUSG000000000031  1.488681  0.000000  0.000000  15.46552   4.290716   0.5620887  1.105509
ENSMUSG000000000037  47.637787  6.771444  0.000000  0.000000  17.162865  3.9346207  31.507014
ENSMUSG000000000049  41.683064  25.731487  27.205900  36.87931   3.064797   28.6665222  6.080301

```

```
class(fa_expr_norm)
```

```
[1] "data.frame"
```

```

fa_meta  <- read.csv("fa_meta.csv",sep=",", row.names = 1, header = TRUE)
fa_meta <- fa_meta[,-c(3,4)]
fa_meta

```

	sampleName	condition
normal_1	normal_1	normal
normal_2	normal_2	normal
normal_3	normal_3	normal
day1_1	day1_1	day1
day1_2	day1_2	day1
day1_3	day1_3	day1
day2_1	day2_1	day2
day2_2	day2_2	day2
day2_3	day2_3	day2
day3_1	day3_1	day3
day3_2	day3_2	day3
day3_3	day3_3	day3
day7_1	day7_1	day7
day7_2	day7_2	day7
day7_3	day7_3	day7
day14_1	day14_1	day14
day14_2	day14_2	day14
day14_3	day14_3	day14

```

fa_annotation <- read.csv("fa_annotations.csv",sep=";", row.names = 1, header = TRUE)
head(fa_annotation)

```

	external_gene_name	chromosome_name	start_position	end_position	strand
ENSMUSG000000000001	Gnai3	3	108014596	108053462	-1
ENSMUSG000000000028	Cdc45	16	18599197	18630737	-1
ENSMUSG000000000037	Scml2	X	159865521	160041209	1
ENSMUSG000000000049	Apoh	11	108234180	108305222	1
ENSMUSG000000000056	Narf	11	121128079	121146682	1
ENSMUSG000000000058	Cav2	6	17281184	17289114	1

```

fa_trait <- read.csv("fa_trait.csv",sep=",", row.names = 1, header = TRUE)
fa_trait

```

	control	day1	day2	day3	day7	day14
normal_1	1	0	0	0	0	0
normal_2	1	0	0	0	0	0

normal_3	1	0	0	0	0	0
day1_1	0	1	0	0	0	0
day1_2	0	1	0	0	0	0
day1_3	0	1	0	0	0	0
day2_1	0	0	1	0	0	0
day2_3	0	0	1	0	0	0
day3_1	0	0	0	1	0	0
day3_2	0	0	0	1	0	0
day3_3	0	0	0	1	0	0
day7_1	0	0	0	0	1	0
day7_2	0	0	0	0	1	0
day7_3	0	0	0	0	1	0
day14_1	0	0	0	0	0	1
day14_2	0	0	0	0	0	1
day14_3	0	0	0	0	0	1

```
## Classer suivant ordre colonne Name de fa_meta
sample_order <- order(fa_meta$sampleName)
fa_meta <- fa_meta[sample_order, ]

## Ordre des echantillons de data.table idem fa_meta table
fa_expr_norm <- fa_expr_norm[, row.names(fa_meta)]
head(fa_expr_norm)
```

	day1_1	day1_2	day1_3	day14_1	day14_2	day14_3	day2_1	...
ENSMUSG00000000000001	2711.22417	1094.745589	1331.5880610	728.668808	603.846446	646.124076	1185.658698	1233.000000
ENSMUSG00000000000003	0.00000	0.000000	0.0000000	0.000000	0.000000	0.000000	0.000000	0.000000
ENSMUSG00000000000028	42.82759	13.485108	21.9214582	16.244209	10.802262	33.839559	166.379146	23.000000
ENSMUSG00000000000031	15.46552	4.290716	0.5620887	1.160301	0.000000	0.000000	1.105509	0.000000
ENSMUSG00000000000037	0.00000	17.162865	3.9346207	6.961804	2.160452	1.057486	31.507014	0.000000
ENSMUSG000000000049	36.87931	3.064797	28.6665222	30.167817	2.160452	59.219228	6.080301	14.000000

fa\_meta

	sampleName	condition
day1_1	day1_1	day1
day1_2	day1_2	day1
day1_3	day1_3	day1
day14_1	day14_1	day14
day14_2	day14_2	day14
day14_3	day14_3	day14
day2_1	day2_1	day2
day2_2	day2_2	day2
day2_3	day2_3	day2
day3_1	day3_1	day3
day3_2	day3_2	day3
day3_3	day3_3	day3
day7_1	day7_1	day7
day7_2	day7_2	day7
day7_3	day7_3	day7
normal_1	normal_1	normal
normal_2	normal_2	normal
normal_3	normal_3	normal

## Filtres

Je filtre et je supprime tous les gènes non exprimés, les NA et je supprime tous les gènes qui ne sont pas exprimés dans au moins 9 des 18 conditions (expression > 1 TPM dans 9)

Puis filtre avec le coefficient de variation > 0,75.

```
dim(fa_expr_norm)
```

```
[1] 46679      18
```

```
#voir le nombre de lignes avec zero counts
rs <- rowSums(fa_expr_norm)
nbgenes_at_zeros <- length(which(rs==0))
nbgenes_at_zeros
```

```
[1] 11457
```

```
#Je supprime les lignes avec zero counts
fa_expr_norm <- fa_expr_norm[rowSums(fa_expr_norm[, -1])>0, ]
dim(fa_expr_norm)
```

```
[1] 35107      18
```

```
#Je supprime les NA
library(tidyr)
fa_expr_norm <- fa_expr_norm %>% drop_na()
dim(fa_expr_norm)
```

```
[1] 35107      18
```

```
#Je ne garde que les gènes où il y a moins de 9 échantillons avec des counts supérieurs ou égaux à 1.

#fa_969 <- fa_expr_norm[rowSums((fa_expr_norm[, -1])>=1) >= 9, ]
#dim(fa_969)

nbexpr <- apply(fa_expr_norm, 1, function(x){length(which(x>=1))})
isexpr <- which(nbexpr>=9)
fa_filtre <- fa_expr_norm[isexpr,]
dim(fa_filtre)
```

```
[1] 24608      18
```

Je filtre avec le coefficient de variation > 0.75 et je **TRANSPOSE** la count table

```
#Je calcul le cv par gène
gene_mean <- apply(fa_filtre, 1, mean)
gene_sd <- apply(fa_filtre, 1, sd)
gene_cv <- gene_sd / gene_mean

#Je filtre les gènes avec un cv > 0.75
fa_cv <- fa_expr_norm[gene_cv > 0.75, ]
dim(fa_cv)
```

```
[1] 13899    18
```

```
#####TRANSPOSER LA COUNT TABLE#####
fa_cv <- t(fa_cv)
```

```
gsg = goodSamplesGenes(fa_cv, verbose = 3);
```

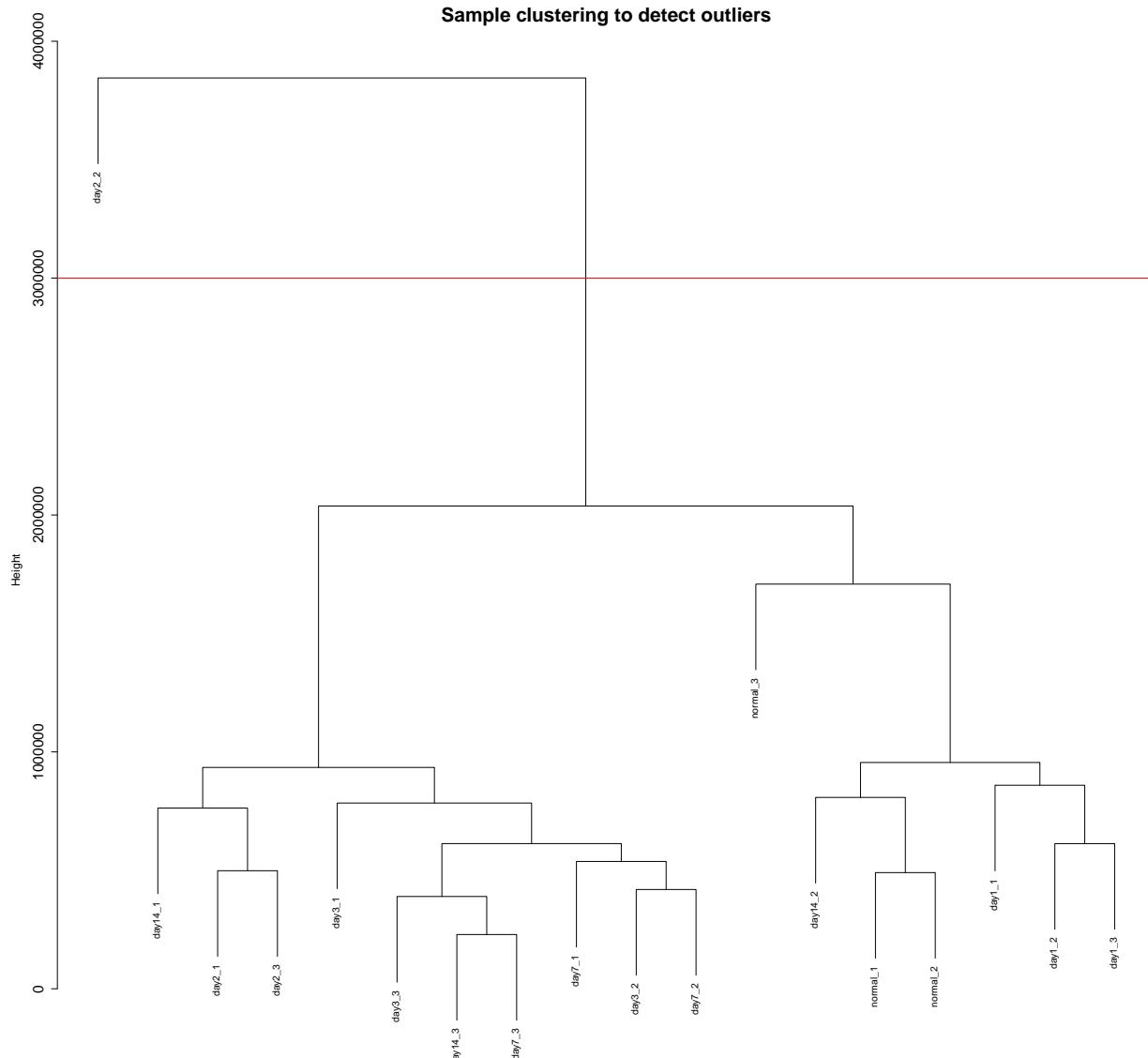
```
Flagging genes and samples with too many missing values...
..step 1
```

```
gsg$allOK
```

```
[1] TRUE
```

### Arbre permettant de détecter les valeurs abérrantes

```
sampleTree = hclust(dist(fa_cv), method = "average");
# Plot the sample tree: Open a graphic output window of size 12 by 10 inches
# The user should change the dimensions if the window is too large or too small.
options(repr.plot.width = 12, repr.plot.height = 10)
plot(sampleTree, main = "Sample clustering to detect outliers", sub="", xlab="",
      cex.lab = 1.2, cex.axis = 1.5, cex.main = 2)
# Plot a line to show the cut
abline(h = 3000000, col = "red");
```



On va couper le cluster pour retirer l'échantillon aberrant

```
# Determine cluster under the line
clust = cutreeStatic(sampleTree, cutHeight = 3000000, minSize = 10)
table(clust)
```

```
clust
0 1
1 17
```

```
# clust 1 contains the samples we want to keep.
keepSamples = (clust==1)
datExpr = fa_cv[keepSamples, ]
```

```

nGenes = ncol(datExpr)
nSamples = nrow(datExpr)
#head(datExpr)
#class(datExpr)
datExpr <- as.data.frame(datExpr)
#head(datExpr)
# datExpr_view <- datExpr[,1:10]
# datExpr_view

```

## Import trait table

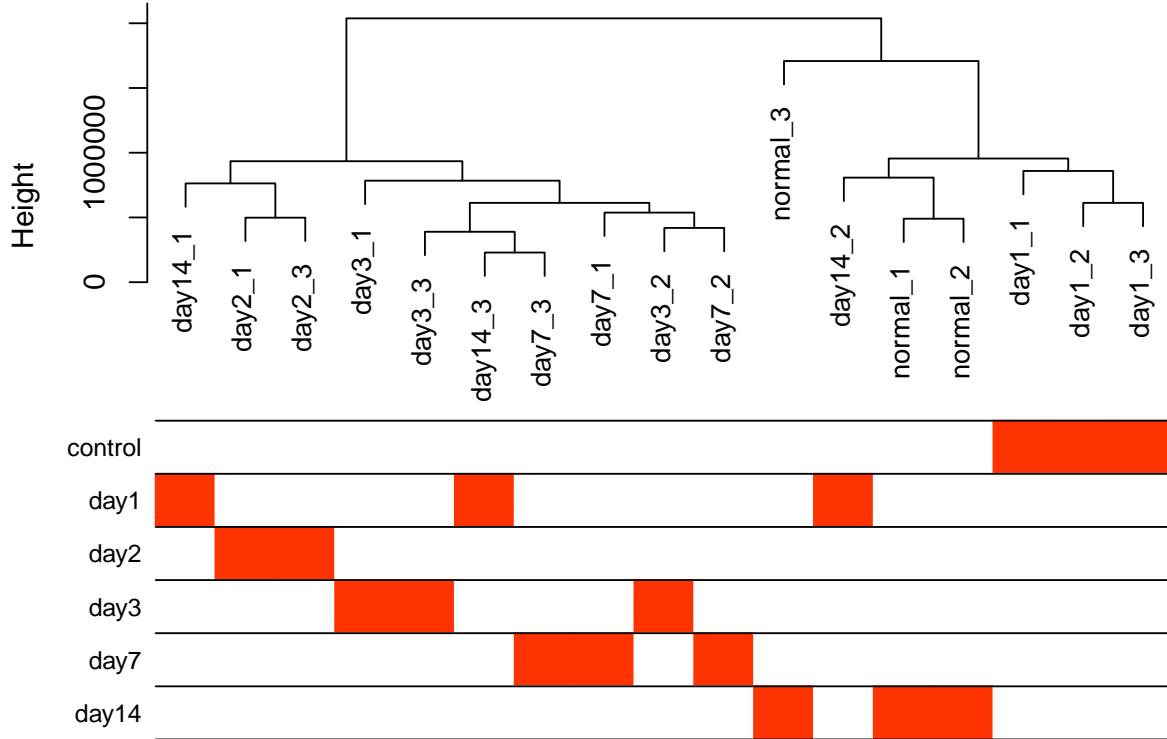
Liaison échantillons et mesures de phénotype avec la table “trait” (analyse binaire)

```

# Re-cluster samples
sampleTree2 = hclust(dist(datExpr), method = "average")
# Convert traits to a color representation: white means low, red means high, grey means missing entry
traitColors = numbers2colors(fa_trait, signed = FALSE);
# Plot the sample dendrogram and the colors underneath.
options(repr.plot.width = 15, repr.plot.height = 12)
plotDendroAndColors(sampleTree2, traitColors,
                     groupLabels = names(fa_trait),
                     main = "Sample dendrogram and trait heatmap")

```

**Sample dendrogram and trait heatmap**



```
# Allow multi-threading within WGCNA. This helps speed up certain calculations.
# At present this call is necessary for the code to work.
# Any error here may be ignored but you may want to update WGCNA if you see one.
# See note above.
allowWGCNAThreads()
```

Allowing multi-threading with up to 56 threads.

```
# Load the data saved in the first part
lnames = load(file = "fa-dataInput.RData");
#The variable lnames contains the names of loaded variables.
lnames
```

```
[1] "datExpr"  "fa_trait"
```

Détection des modules et construction du réseau WGCNA

```
# Choose a set of soft-thresholding powers
powers = c(c(1:10), seq(from = 12, to = 20, by = 2)) #on prend de 12 à 20, 2 en 2
# Call the network topology analysis function
sft = pickSoftThreshold(datExpr, powerVector = powers, verbose = 5)
```

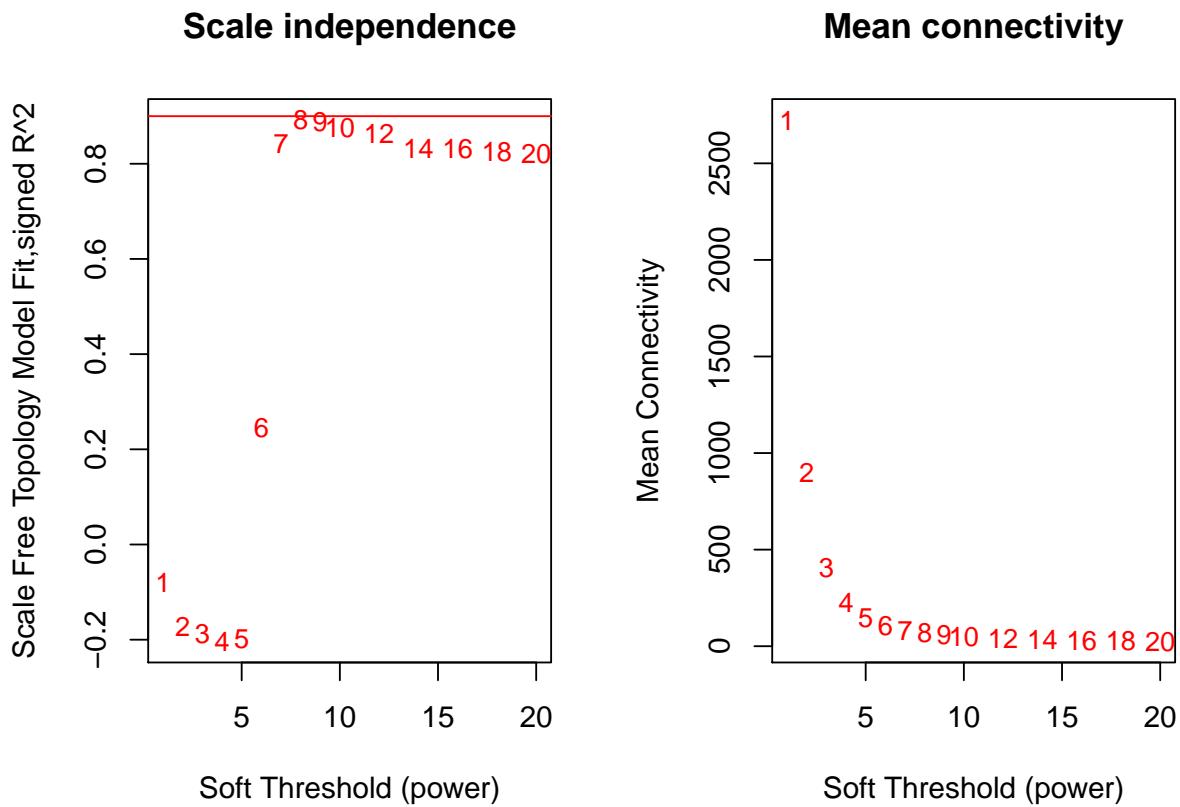
```
pickSoftThreshold: will use block size 3218.
pickSoftThreshold: calculating connectivity for given powers...
..working on genes 1 through 3218 of 13899
..working on genes 3219 through 6436 of 13899
..working on genes 6437 through 9654 of 13899
..working on genes 9655 through 12872 of 13899
..working on genes 12873 through 13899 of 13899
Power SFT.R.sq slope truncated.R.sq mean.k. median.k. max.k.
1      1  0.0801   3.010      0.7320  2720.0  2730.00  3590
2      2  0.1710   3.880      0.4230  899.0   886.00  1410
3      3  0.1870   3.610      0.0882  406.0   399.00   714
4      4  0.2040   2.900     -0.0127  228.0   212.00   429
5      5  0.1980   0.439      0.7670  148.0   125.00   329
6      6  0.2450  -0.394      0.3250  107.0   79.20   293
7      7  0.8430  -0.848      0.7980  82.9    53.00   269
8      8  0.8920  -0.914      0.8850  67.5    36.80   252
9      9  0.8880  -0.929      0.8930  57.0    26.30   239
10     10 0.8760  -0.933      0.8750  49.5    19.20   229
11     12 0.8640  -0.912      0.8800  39.6    10.80   213
12     14 0.8320  -0.909      0.8230  33.4    6.43    203
13     16 0.8310  -0.888      0.8280  29.2    3.97    194
14     18 0.8260  -0.869      0.8310  26.1    2.52    188
15     20 0.8200  -0.857      0.8250  23.8    1.66    183
```

```
# Plot the results:
par(mfrow = c(1, 2));
options(repr.plot.width = 14, repr.plot.height = 10);
# Scale-free topology fit index as a function of the soft-thresholding power
plot(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
      xlab = "Soft Threshold (power)", ylab = "Scale Free Topology Model Fit,signed R^2", type = "n",
```

```

main = paste("Scale independence");
text(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
      labels = powers, cex = 0.9, col = "red");
# this line corresponds to using an R^2 cut-off of h
abline(h = 0.90, col = "red")
# Mean connectivity as a function of the soft-thresholding power
plot(sft$fitIndices[,1], sft$fitIndices[,5],
      xlab = "Soft Threshold (power)", ylab = "Mean Connectivity", type = "n",
      main = paste("Mean connectivity"))
text(sft$fitIndices[,1], sft$fitIndices[,5], labels = powers, cex = 0.9, col = "red")

```



On voit qu'à partir de puissance 7 on atteint le seuil donc pas besoin d'aller jusqu'à puissance 20 on va mettre matrice de corrélation à la puissance 8

Graph 2: quand on augmente la puissance on perd des corrélations

## Adjacency matrix

Ci dessous, je choisis 8 comme puissance la plus basse qui construit la scale free topology. Ensuite, la fonction de génère des modules de taille 30, et fusionne les modules similaires à plus de 25% et enregistre la matrice de chevauchement topologique dans un objet.

```
#coupe le jeu de données en differents blocks pour eviter methode de calculation plus lourde
net = blockwiseModules(datExpr, power = 8,
                       TOMType = "signed", minModuleSize = 30,
                       reassignThreshold = 0.5, mergeCutHeight = 0.25,
                       numericLabels = TRUE, pamRespectsDendro = FALSE,
                       saveTOMs = TRUE, nThreads = 8,
                       saveTOMfileBase = "fa_TOM",
                       verbose = 3)
```

```
Calculating module eigengenes block-wise from all genes
Flagging genes and samples with too many missing values...
..step 1
..Excluding 128 genes from the calculation due to too many missing samples or zero variance.
..step 2
....pre-clustering genes to determine blocks..
Projective K-means:
..k-means clustering..
..merging smaller clusters...
Block sizes:
gBlocks
    1     2     3
4789  4702  4280
..Working on block 1 .
    TOM calculation: adjacency..
    ..will use 8 parallel threads.
    Fraction of slow calculations: 0.000000
    ..connectivity..
    ..matrix multiplication (system BLAS)..
    ..normalization..
    ..done.
    ..saving TOM for block 1 into file fa_TOM-block.1.RData
....clustering..
....detecting modules..
....calculating module eigengenes..
....checking kME in modules..
    ..removing 4 genes from module 1 because their KME is too low.
    ..removing 17 genes from module 2 because their KME is too low.
    ..removing 13 genes from module 3 because their KME is too low.
    ..removing 15 genes from module 4 because their KME is too low.
    ..removing 1 genes from module 5 because their KME is too low.
    ..removing 2 genes from module 6 because their KME is too low.
    ..removing 1 genes from module 8 because their KME is too low.
..Working on block 2 .
    TOM calculation: adjacency..
    ..will use 8 parallel threads.
    Fraction of slow calculations: 0.000000
    ..connectivity..
    ..matrix multiplication (system BLAS)..
    ..normalization..
    ..done.
    ..saving TOM for block 2 into file fa_TOM-block.2.RData
....clustering..
....detecting modules..
```

```

....calculating module eigengenes..
....checking kME in modules..
    ..removing 12 genes from module 2 because their KME is too low.
    ..removing 1 genes from module 3 because their KME is too low.
    ..removing 10 genes from module 4 because their KME is too low.
    ..removing 21 genes from module 5 because their KME is too low.
    ..removing 8 genes from module 6 because their KME is too low.
..Working on block 3 .
    TOM calculation: adjacency..
    ..will use 8 parallel threads.
    Fraction of slow calculations: 0.000000
    ..connectivity..
    ..matrix multiplication (system BLAS)..
    ..normalization..
    ..done.
    ..saving TOM for block 3 into file fa_TOM-block.3.RData
....clustering..
....detecting modules..
....calculating module eigengenes..
....checking kME in modules..
    ..removing 5 genes from module 1 because their KME is too low.
    ..removing 22 genes from module 2 because their KME is too low.
    ..removing 4 genes from module 3 because their KME is too low.
    ..removing 3 genes from module 4 because their KME is too low.
    ..removing 10 genes from module 5 because their KME is too low.
    ..removing 1 genes from module 6 because their KME is too low.
    ..reassigning 146 genes from module 1 to modules with higher KME.
    ..reassigning 109 genes from module 2 to modules with higher KME.
    ..reassigning 99 genes from module 3 to modules with higher KME.
    ..reassigning 107 genes from module 4 to modules with higher KME.
    ..reassigning 158 genes from module 5 to modules with higher KME.
    ..reassigning 128 genes from module 6 to modules with higher KME.
    ..reassigning 37 genes from module 7 to modules with higher KME.
    ..reassigning 26 genes from module 8 to modules with higher KME.
    ..reassigning 4 genes from module 9 to modules with higher KME.
    ..reassigning 2 genes from module 10 to modules with higher KME.
    ..reassigning 195 genes from module 11 to modules with higher KME.
    ..reassigning 139 genes from module 12 to modules with higher KME.
    ..reassigning 143 genes from module 13 to modules with higher KME.
    ..reassigning 129 genes from module 14 to modules with higher KME.
    ..reassigning 94 genes from module 15 to modules with higher KME.
    ..reassigning 45 genes from module 16 to modules with higher KME.
    ..reassigning 5 genes from module 17 to modules with higher KME.
    ..reassigning 5 genes from module 18 to modules with higher KME.
    ..reassigning 2 genes from module 19 to modules with higher KME.
    ..reassigning 122 genes from module 20 to modules with higher KME.
    ..reassigning 129 genes from module 21 to modules with higher KME.
    ..reassigning 87 genes from module 22 to modules with higher KME.
    ..reassigning 69 genes from module 23 to modules with higher KME.
    ..reassigning 129 genes from module 24 to modules with higher KME.
    ..reassigning 55 genes from module 25 to modules with higher KME.
..merging modules that are too close..
    mergeCloseModules: Merging modules whose distance is less than 0.25
    Calculating new MEs...

```

Nombre et taille des modules

```
table(net$colors)
```

Module	Nombre de gènes
0	299
1	802
2	796
3	780
4	755
5	735
6	730
7	714
8	696
9	687
10	684
11	674
12	672
13	667
14	638
15	613
16	571
17	521
18	504
19	397
20	186
21	179
22	132
23	129
24	128
25	109

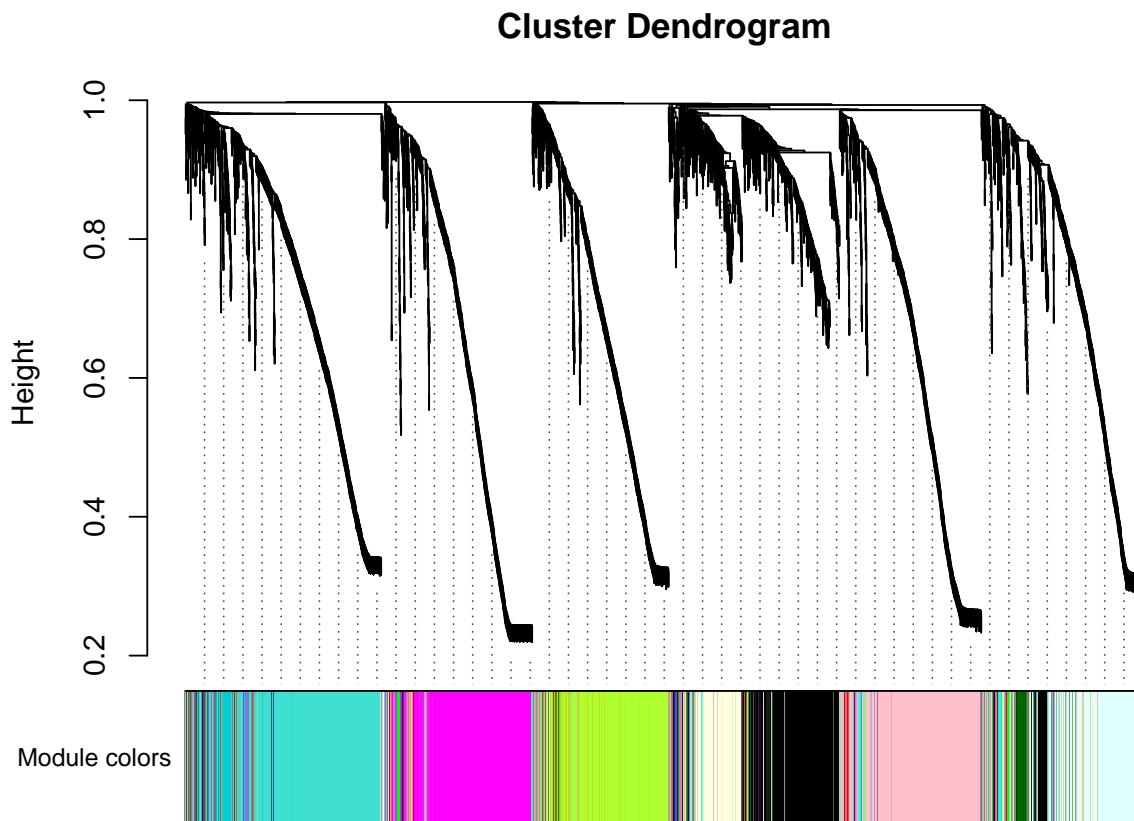
Il y a **26** modules

## Cluster Dendrogram

Ci-dessous la représentation des modules et du clustering des gènes

```
# Convert labels to colors for plotting
mergedColors = labels2colors(net$colors)
#mergedColors

# Plot the dendrogram and the module colors underneath
plotDendroAndColors(net$dendrograms[[1]], mergedColors[net$blockGenes[[1]]],
                     "Module colors",
                     dendroLabels = FALSE, hang = 0.03,
                     addGuide = TRUE, guideHang = 0.05)
```



Sauvegarde des résultats en .RData

```
#on transforme en couleurs

moduleLabels = net$colors
moduleColors = labels2colors(net$colors)
MEs = net$MEs;
head(MEs)
```

	ME10	ME13	ME18	ME9	ME1	ME19	ME12	ME23
day1_1	-0.07238678	-0.05224553	-0.04349821	-0.05329046	-0.07830409	0.4157200	0.96798795	0.74174292
day1_2	-0.04933208	0.96786593	-0.01574374	-0.05890943	-0.04434104	0.3920358	-0.03757649	-0.07550500
day1_3	-0.04975890	-0.01365559	0.96835907	-0.02935999	-0.03914447	0.3722032	-0.02576257	-0.07280541
day14_1	-0.03771846	-0.07454078	-0.05989101	-0.08794690	-0.08017217	-0.2358184	-0.05945834	-0.12446948
day14_2	-0.08620551	-0.08277742	-0.06936665	-0.07724318	-0.09439856	-0.2470120	-0.06524764	-0.08311593
day14_3	-0.03781602	-0.06884687	-0.07689919	-0.06888207	-0.05307051	-0.2408391	-0.07299399	-0.14055045
	ME7	ME8	ME0					
day1_1	-0.1152950	-0.06506602	-0.01723925					
day1_2	-0.1740444	-0.07430235	-0.28786238					
day1_3	-0.1351143	-0.05630494	-0.17468333					
day14_1	0.1162694	-0.05588556	0.17096996					
day14_2	0.1866544	-0.04035597	0.35816433					
day14_3	-0.1289810	-0.06817616	0.10376559					

```
geneTree = net$dendrograms[[1]];
save(MEs, moduleLabels, moduleColors, geneTree,
      file = "Transcriptomique-networkConstruction-auto.RData")
```

```
lnames = load(file = "fa-dataInput.RData");
#The variable lnames contains the names of loaded variables.
lnames
```

```
[1] "datExpr"   "fa_trait"

# Load network data saved in the second part.
lnames = load(file = "Transcriptomique-networkConstruction-auto.RData");
lnames
```

```
[1] "MEs"           "moduleLabels" "moduleColors" "geneTree"
```

Quantification des associations module-trait

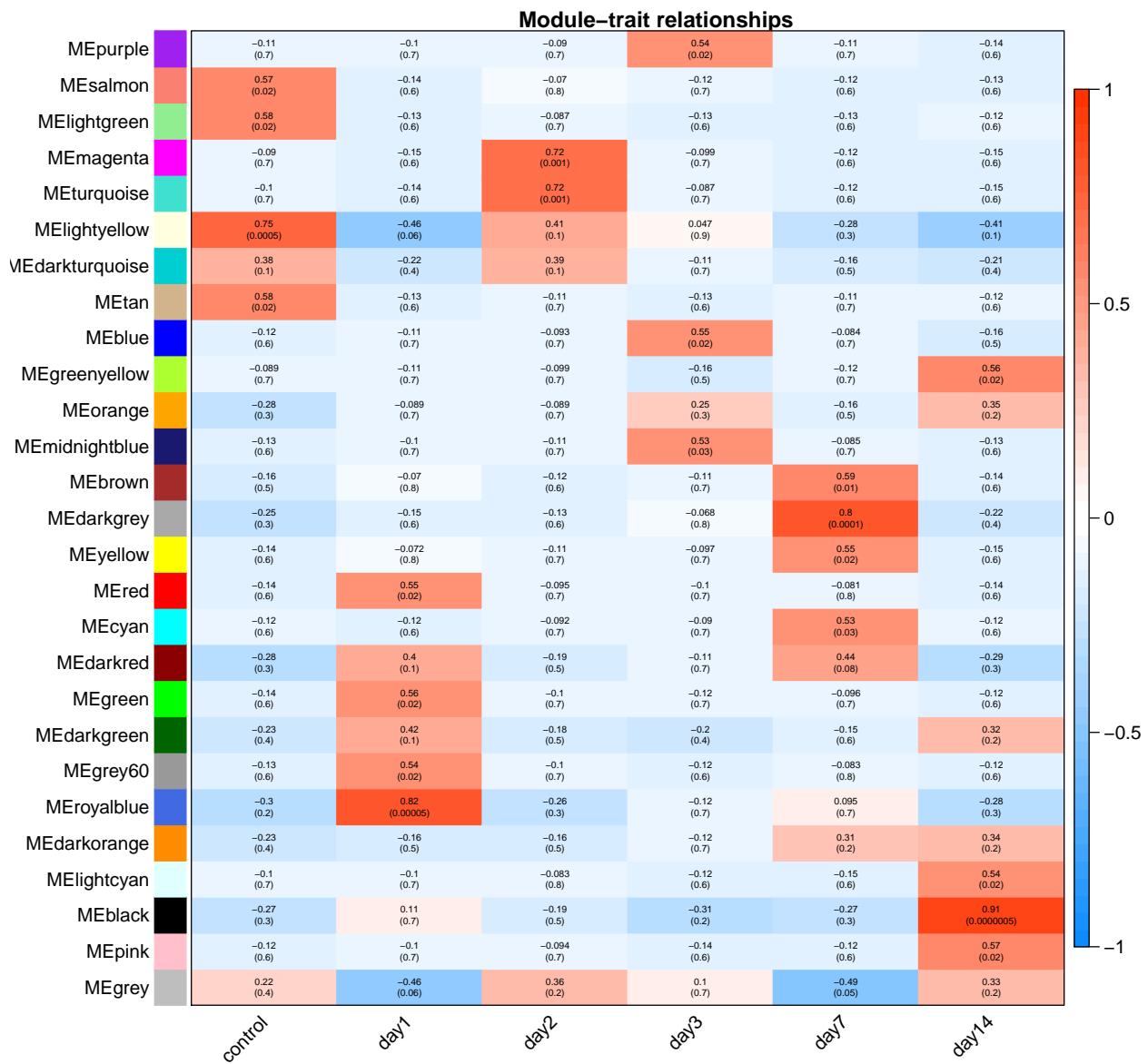
Les modules qui sont significativement associés aux traits cliniques sont mesurés. Nous avons déjà un profil de synthèse calculé (eigengene) pour chaque module, donc nous corrélons simplement les eigengènes avec des traits phénotypiques et recherchons les associations les plus significatives:

```
# Define numbers of genes and samples
nGenes = ncol(datExpr);
nSamples = nrow(datExpr);
# Recalculate MEs with color labels
MEs0 = moduleEigengenes(datExpr, moduleColors)$eigengenes
MEs = orderMEs(MEs0)
moduleTraitCor = cor(MEs, fa_trait, use = "p");
moduleTraitPvalue = corPvalueStudent(moduleTraitCor, nSamples);
```

## Heatmap relation module/trait

Représentation de chaque module eigengene et son coefficient de corrélation.

```
# Will display correlations and their p-values
textMatrix = paste(signif(moduleTraitCor, 2), "\n",
                   signif(moduleTraitPvalue, 1), ")",
                   sep = "");
dim(textMatrix) = dim(moduleTraitCor)
par(mar = c(6, 8, 1, 1));
# Display the correlation values within a heatmap plot
labeledHeatmap(Matrix = moduleTraitCor,
                xLabels = names(fa_trait),
                yLabels = names(MEs),
                ySymbols = names(MEs),
                colorLabels = FALSE,
                colors = blueWhiteRed(50),
                textMatrix = textMatrix,
                setStdMargins = FALSE,
                cex.text = 0.5,
                zlim = c(-1,1),
                main = paste("Module-trait relationships"))
```

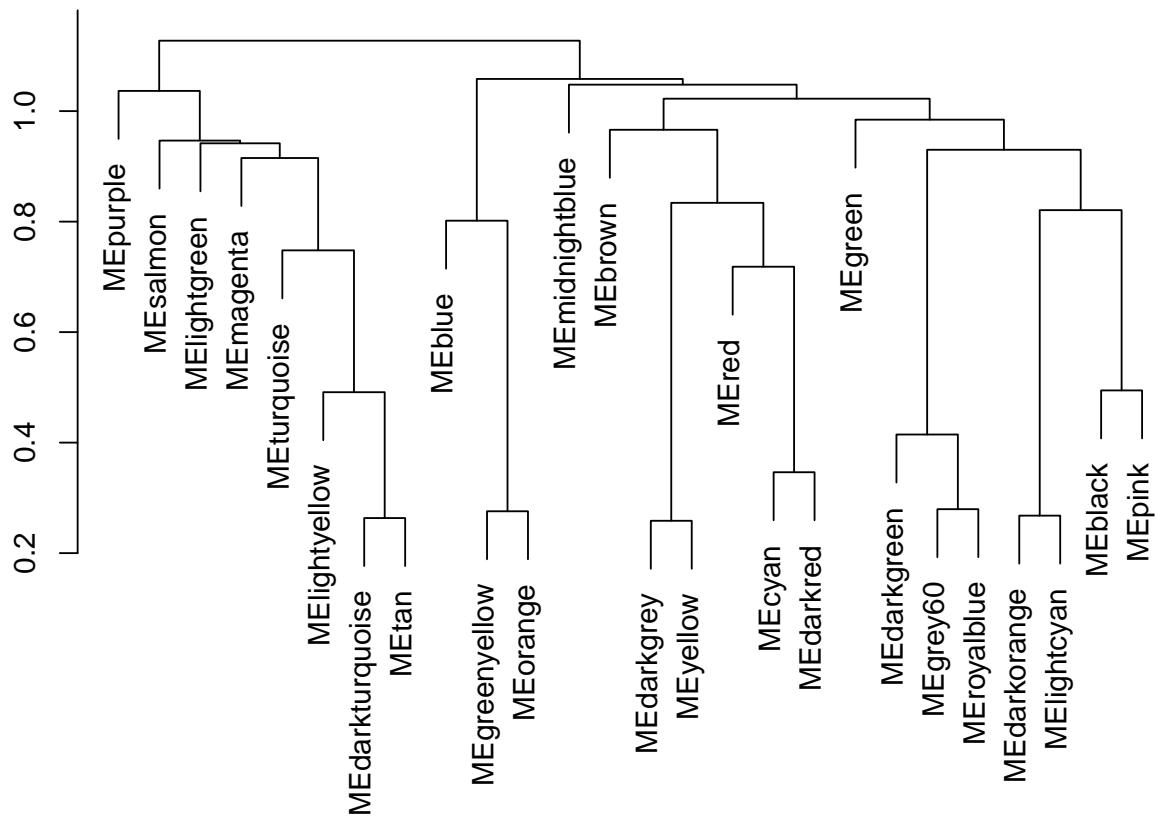


Correlation entre la matrice de eigengene MEs et la matrice de datTraits On peut voir les fortes corrélations rouge foncé ou bleu foncé

## Eigengene view

```
plotEigengeneNetworks(MEs, "Eigengene dendrogram", marDendro = c(0,4,2,0), plotHeatmaps = FALSE)
```

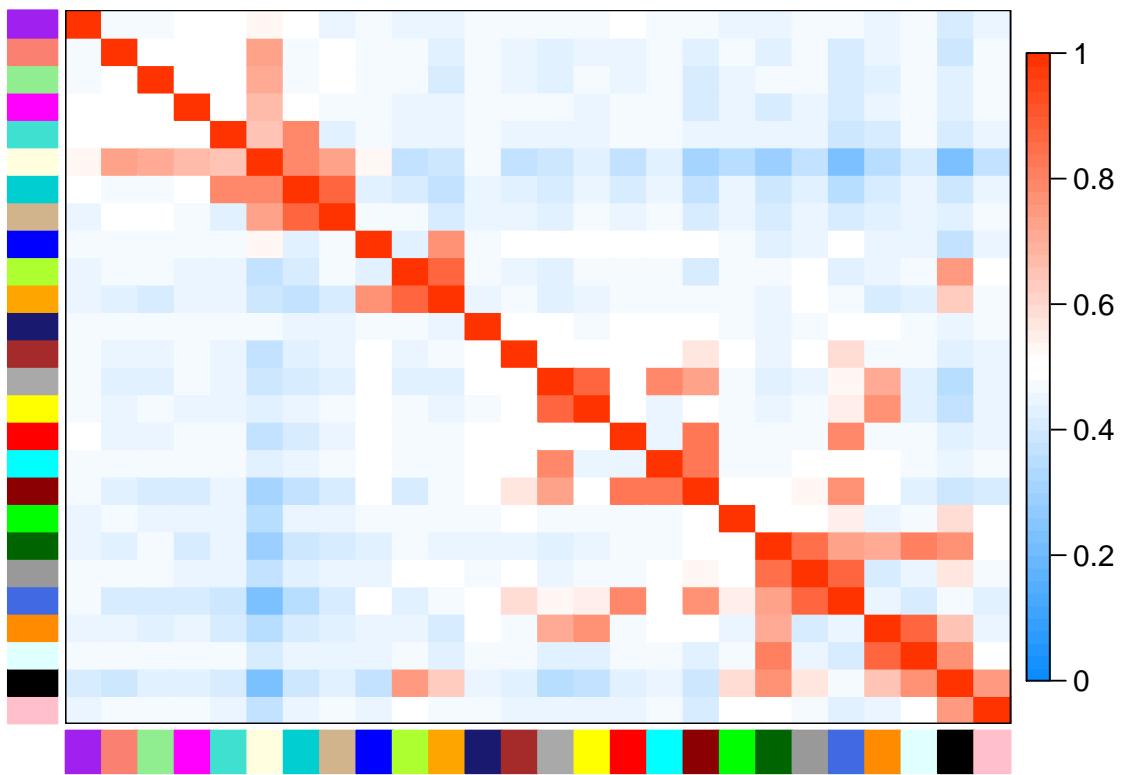
## Eigengene dendrogram



Heatmap matrix

```
# Plot the heatmap matrix (note: this plot will overwrite the dendrogram plot)
par(cex = 1.0)
plotEigengeneNetworks(MEs, "Eigengene adjacency heatmap", marHeatmap = c(3,4,2,2), plotDendograms = FALSE)
```

### Eigengene adjacency heatmap



Visualize, analyze the network and superimpose the proteomics data on it.

Colorez dans le réseau choisi les noeuds en fonction des données de protéomiques avec un gradient de couleur correspondant au fold-change des données de protéomique.

### Export Cytoscape

```
# Recalculate topological overlap if needed
TOM=TOMsimilarityFromExpr(datExpr, power=8)
```

```
TOM calculation: adjacency..
..will use 56 parallel threads.
Fraction of slow calculations: 0.000000
..connectivity..
..matrix multiplication (system BLAS)..
..normalization..
..done.
```

```
# Read in the annotation file
#annot=read.csv(file="data/GeneAnnotation.csv")
# Select modules
modules=c("darkgrey", "grey")
```

```

# Select module probes
probes=colnames(datExpr)
inModule=is.finite(match(moduleColors, modules))
modProbes=probes[inModule]
#modGenes=annot$gene_symbol[match(modProbes, annot$substanceBXH)]
# Select the corresponding Topological Overlap
modTOM=TOM[inModule, inModule]
dimnames(modTOM)=list(modProbes, modProbes)
# Export the network into edge and node list files Cytoscape can read
cyt=exportNetworkToCytoscape(modTOM,
                           edgeFile=paste("CytoscapeInput-edges-", paste(modules, collapse="-"), ".txt", sep=""),
                           nodeFile=paste("CytoscapeInput-nodes-", paste(modules, collapse="-"), ".txt", sep=""),
                           weighted=TRUE,
                           threshold=0.5,
                           nodeNames=modProbes)
                           #altNodeName

```

Je ne comprends pas pourquoi mes exports nodes et edges sont vides donc pas possible de passer à Cytoscape

## Remise du rapport

Vous fournirez un rapport au format pdf généré à partir d'un Rmd (déposez-vous impérativement les 2 fichiers, Rmd et pdf, avec comme nom de fichier "NOM-PRENOM\_evaluation-m6-2021" + .Rmd ou .pdf dans le dossier /shared/projects/dubii2021//m6-bioinfo-integr/mini-projet/)

## Session info

```

#### Session info ####
sessionInfo()

R version 4.0.3 (2020-10-10)
Platform: x86_64-conda-linux-gnu (64-bit)
Running under: CentOS Linux 7 (Core)

Matrix products: default
BLAS/LAPACK: /shared/ifbstor1/software/miniconda/envs/r-4.0.3/lib/libopenblas-r0.3.10.so

locale:
[1] LC_CTYPE=en_US.UTF-8          LC_NUMERIC=C                  LC_TIME=en_US.UTF-8          LC_COLLATE=en_US.UTF-8
[11] LC_MEASUREMENT=en_US.UTF-8   LC_IDENTIFICATION=C

attached base packages:
[1] parallel stats4   stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] tidyverse_1.1.3              WGCNA_1.70-3                fastcluster_1.1.25          dynamicTreeCut_
[11] Biobase_2.50.0               MatrixGenerics_1.2.1         matrixStats_0.58.0          GenomicRanges_
[21] gprofiler2_0.2.0             factoextra_1.0.7            ggplot2_3.3.3               FactoMineR_2.4

loaded via a namespace (and not attached):

```

```
[1] colorspace_2.0-1           ellipsis_0.3.2          htmlTable_2.2.1          corpcor_1.6.9           XVector_2.10.0
[13] RSpectra_0.16-0          fansi_0.4.2              codetools_0.2-18        splines_4.0.3           leaps_3.0
[25] cluster_2.1.1            GO.db_3.12.1             png_0.1-7               compiler_4.0.3          httr_1.4.0
[37] igraph_1.2.6              gtable_0.3.0             glue_1.4.2              GenomeInfoDbData_1.2.4 reshape_0.1
[49] XML_3.99-0.6              zlibbioc_1.36.0          scales_1.1.1            RColorBrewer_1.1-2      yaml_2.2.0
[61] genefilter_1.72.1          foreach_1.5.1            checkmate_2.0.0         BiocParallel_1.24.1     rlang_0.4.1
[73] tidyselect_1.1.1          plyr_1.8.6               magrittr_2.0.1           R6_2.5.0                generics_0.0.2
[85] survival_3.2-10           scatterplot3d_0.3-41    RCurl_1.98-1.3          crayon_1.4.1            rARPA_0.1.0
[97] data.table_1.14.0          blob_1.2.1               digest_0.6.27           flashClust_1.01-2      xtable_2.2.0
```