Mapping the HDL proteome in Metabolic Syndrome through label-free quantification.

Elvira Márquez Paradas

2023-06-05

The packages required for the execution of this script can be loaded below:

```
# List of packages to be used
packages_cran <- c("tidyverse", "ggplot2", "tidyr", "stats", "factoextra","FactoMineR")
packages_bioconductor <- c("limma", "clusterProfiler", "org.Hs.eg.db", "treemap")

# Check if CRAN packages are installed and, if not, install them.
for (package in packages_cran) {
  if (!requireNamespace(package, quietly = TRUE)) {
    install.packages(package)
  }
}

# Check if the Bioconductor packages are installed and, if not, install them.
if (!requireNamespace("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
}
BiocManager::install(packages_bioconductor)
```

```
## Bioconductor version 3.16 (BiocManager 1.30.20), R 4.2.1 (2022-06-23 ucrt)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   'force = TRUE' to re-install: 'limma' 'clusterProfiler' 'org.Hs.eg.db'
##   'treemap'

## Installation paths not writeable, unable to update packages
##   path: C:/Program Files/R/R-4.2.1/library
##   packages:
##     boot, class, cluster, codetools, foreign, KernSmooth, lattice, MASS,
##     Matrix, mgcv, nlme, nnet, rpart, spatial, survival

## Old packages: 'BiocManager', 'broom', 'bslib', 'curl', 'DEoptimR', 'gargle',
##   'googledrive', 'googlesheets4', 'gprofiler2', 'igraph', 'jsonlite', 'locfit',
##   'mice', 'mvtnorm', 'pkgbuild', 'reticulate', 'rlang', 'robustbase',
##   'scattermore', 'scatterpie', 'testthat', 'usethis', 'vctrs'
```

```
# Load packages
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3

## Warning: package 'ggplot2' was built under R version 4.2.3

## Warning: package 'tibble' was built under R version 4.2.3

## Warning: package 'tidyr' was built under R version 4.2.2

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'forcats' was built under R version 4.2.2

## Warning: package 'lubridate' was built under R version 4.2.2

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(tidyr)
library(stats)
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.2.2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(limma)
```

```
## Warning: package 'limma' was built under R version 4.2.2
```

```
library(FactoMineR)
```

```
## Warning: package 'FactoMineR' was built under R version 4.2.3
```

```
library(clusterProfiler)
```

```
## Warning: package 'clusterProfiler' was built under R version 4.2.2
```

```
##
## clusterProfiler v4.6.2  For help: https://yulab-smu.top/biomedical-knowledge-mining-book/
##
## If you use clusterProfiler in published research, please cite:
## T Wu, E Hu, S Xu, M Chen, P Guo, Z Dai, T Feng, L Zhou, W Tang, L Zhan, X Fu, S Liu, X Bo, and G Yu.
##
## Attaching package: 'clusterProfiler'
##
## The following object is masked from 'package:purrr':
##
##     simplify
##
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(org.Hs.eg.db)
```

```
## Loading required package: AnnotationDbi
```

```
## Warning: package 'AnnotationDbi' was built under R version 4.2.2
```

```
## Loading required package: stats4
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
##
## The following object is masked from 'package:limma':
##
##     plotMA
##
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
##
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union
##
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
##
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min
##
## Loading required package: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
##
## Loading required package: IRanges
## Loading required package: S4Vectors

## Warning: package 'S4Vectors' was built under R version 4.2.2

##
## Attaching package: 'S4Vectors'
##
## The following object is masked from 'package:clusterProfiler':
##
##     rename
##
## The following objects are masked from 'package:lubridate':
##
##     second, second<-
##
## The following objects are masked from 'package:dplyr':
##
##     first, rename
##
## The following object is masked from 'package:tidyr':
##
##     expand
##
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
##
##
## Attaching package: 'IRanges'
##
## The following object is masked from 'package:clusterProfiler':
##
##     slice
##
## The following object is masked from 'package:lubridate':
##
##     %within%
##
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
##
## The following object is masked from 'package:purrr':
##
##     reduce
##
## The following object is masked from 'package:grDevices':
```

```
##
##     windows
##
##
## Attaching package: 'AnnotationDbi'
##
## The following object is masked from 'package:clusterProfiler':
##
##     select
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(treemap)
```

```
## Warning: package 'treemap' was built under R version 4.2.3
```

DATA PREPROCESSING

First, we load the data, select the columns of interest and rename them to facilitate the analysis. We have proteomics data from 17 healthy patients and 20 patients with Metabolic Syndrome. For the analysis, we will use the **LFQ intensity**, which will allow the comparison of the relative abundance of proteins between different samples.

```
patients <- read.table("./data/proteinGroups.txt", header = T, sep = "\t", na.strings = "NaN")
dim(patients)
```

```
## [1] 580 371
```

```
patients <- dplyr::select(patients, select = c("Protein.IDs", "Protein.names", "Gene.names",
                                               matches("^LFQ.intensity."), "Only.identified.by.site", "Reverse

names <- read.table("./data/names.txt", sep = "\n")[,1]
colnames(patients)=names
```

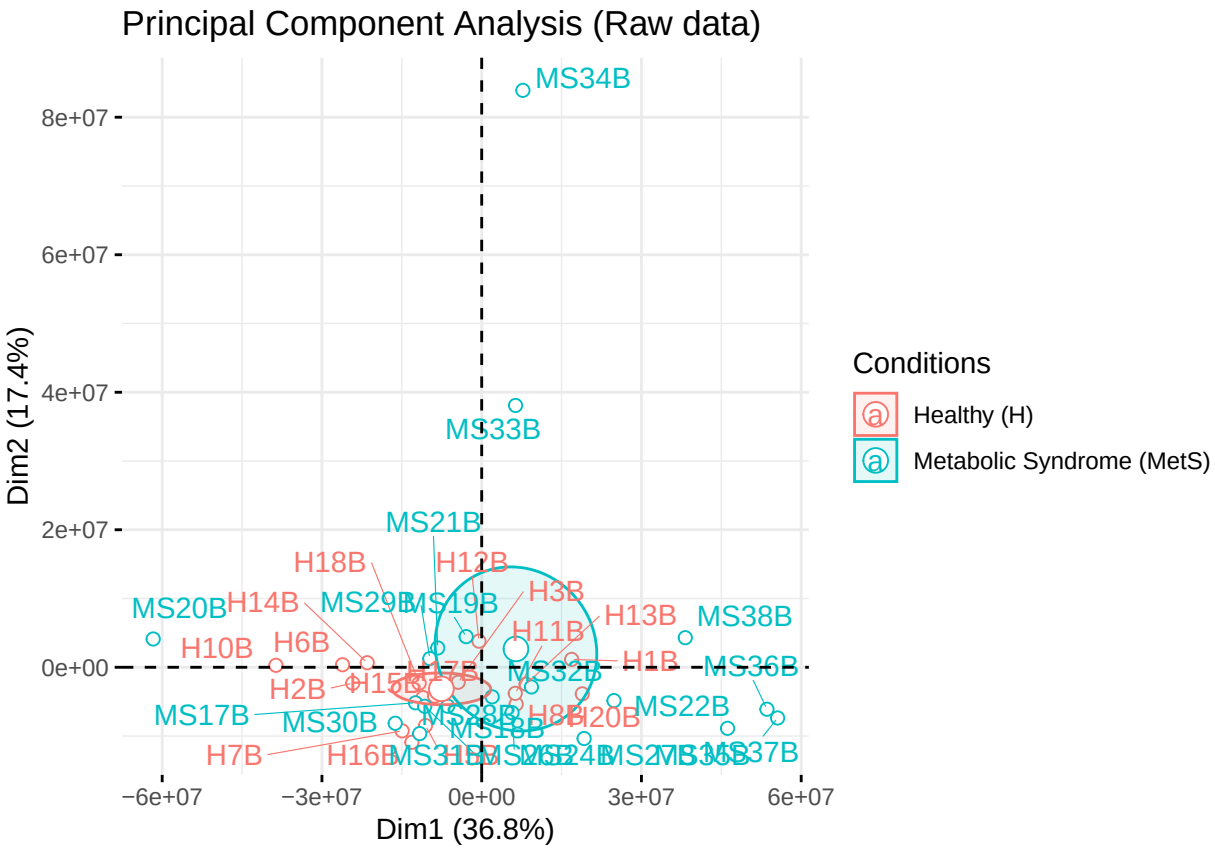
The number of proteins before data pre-processing is 580. Let's check the behavior of the samples before data pre-processing

```
# Reorganization of data
#####
patients.pca= patients[,-c(1:3,41:43)]
pca.data<- data.frame(colnames(patients.pca),t(patients.pca))
colnames(pca.data)[1] <- "Sample"
pca.data[, -1] <- scale(pca.data[, -1], center = T, scale = F)

# Principal Component Analysis
#####
res.pca <- PCA(pca.data[,-1], graph = F,scale.unit = F, ncp = 10 )
res.pca$eig
```

```
##           eigenvalue percentage of variance cumulative percentage of variance
## comp 1  5.494144e+14           36.822243976                             36.82224
## comp 2  2.591205e+14           17.366486773                             54.18873
## comp 3  1.690400e+14           11.329210408                             65.51794
## comp 4  1.325280e+14            8.882142876                             74.40008
## comp 5  9.641057e+13            6.461522151                             80.86161
## comp 6  6.553888e+13            4.392474422                             85.25408
## comp 7  4.819319e+13            3.229950678                             88.48403
## comp 8  3.600409e+13            2.413025920                             90.89706
## comp 9  2.421058e+13            1.622614342                             92.51967
## comp 10 2.058846e+13            1.379856904                             93.89953
## comp 11 1.567178e+13            1.050336908                             94.94987
## comp 12 1.489554e+13            0.998312114                             95.94818
## comp 13 1.184766e+13            0.794040681                             96.74222
## comp 14 6.835870e+12            0.458146090                             97.20036
## comp 15 6.067332e+12            0.406638019                             97.60700
## comp 16 5.176256e+12            0.346917313                             97.95392
## comp 17 4.247458e+12            0.284668388                             98.23859
## comp 18 4.035611e+12            0.270470238                             98.50906
## comp 19 3.281401e+12            0.219922396                             98.72898
## comp 20 3.031766e+12            0.203191656                             98.93217
## comp 21 2.870896e+12            0.192409992                             99.12458
## comp 22 2.453365e+12            0.164426702                             99.28901
## comp 23 2.307116e+12            0.154624944                             99.44363
## comp 24 1.706897e+12            0.114397732                             99.55803
## comp 25 1.265546e+12            0.084818052                             99.64285
## comp 26 1.034471e+12            0.069331192                             99.71218
## comp 27 8.603902e+11            0.057664115                             99.76984
## comp 28 7.808627e+11            0.052334113                             99.82218
## comp 29 5.532908e+11            0.037082046                             99.85926
## comp 30 4.803158e+11            0.032191190                             99.89145
## comp 31 4.139814e+11            0.027745405                             99.91920
## comp 32 3.586273e+11            0.024035520                             99.94323
## comp 33 3.281970e+11            0.021996055                             99.96523
## comp 34 2.031540e+11            0.013615563                             99.97884
## comp 35 1.700448e+11            0.011396552                             99.99024
## comp 36 1.456050e+11            0.009758577                            100.00000
```

```r
# Individual plot
#####
H <- rep("Healthy (H)", 17)
MS <- rep("Metabolic Syndrome (MetS)", 20)
fviz_pca_ind(res.pca, col.ind = c(H,MS),
             pointsize=2, pointshape=21, fill="white",
             repel = TRUE,
             addEllipses = TRUE,ellipse.type = "confidence",
             ellipse.level= 0.95,
             legend.title="Conditions",
             title="Principal Component Analysis (Raw data)",
             show_legend=TRUE,show_guide=TRUE)
```



We replace the 0's by missing values (NA) and transforme the data to logarithmic scale to decrease the variance and homogenize the data.

```
# 0's deletion
#####
patients[patients == 0] <- NA

# Log2 transformation
#####
cols <- names(patients)[grepl("^H|^MS", names(patients))]
patients[cols] <- lapply(patients[cols], log2)
```

We filter the proteins, eliminating reversed proteins, dentified only by site, and potential contaminants. Finally, we remove from the dataset those columns that will not be used again.

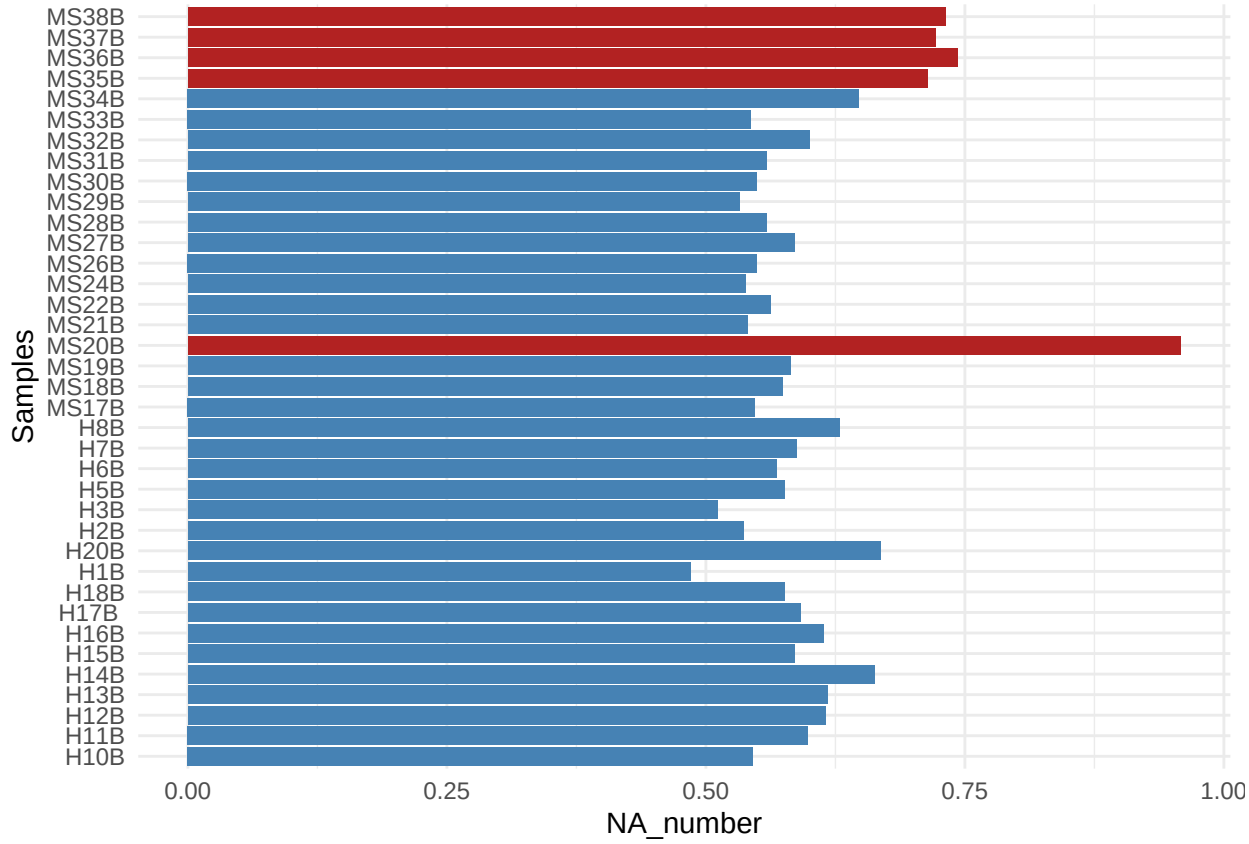
```
patients <- patients %>% filter(Potential.contaminant != "+" & Only.identified.by.site != "+" & Reverse
dim(patients)
```

```
## [1] 528  43
```

```
patients <- patients %>% dplyr::select(-any_of(c("Potential.contaminant","Only.identified.by.site", "Re
data <- patients
```

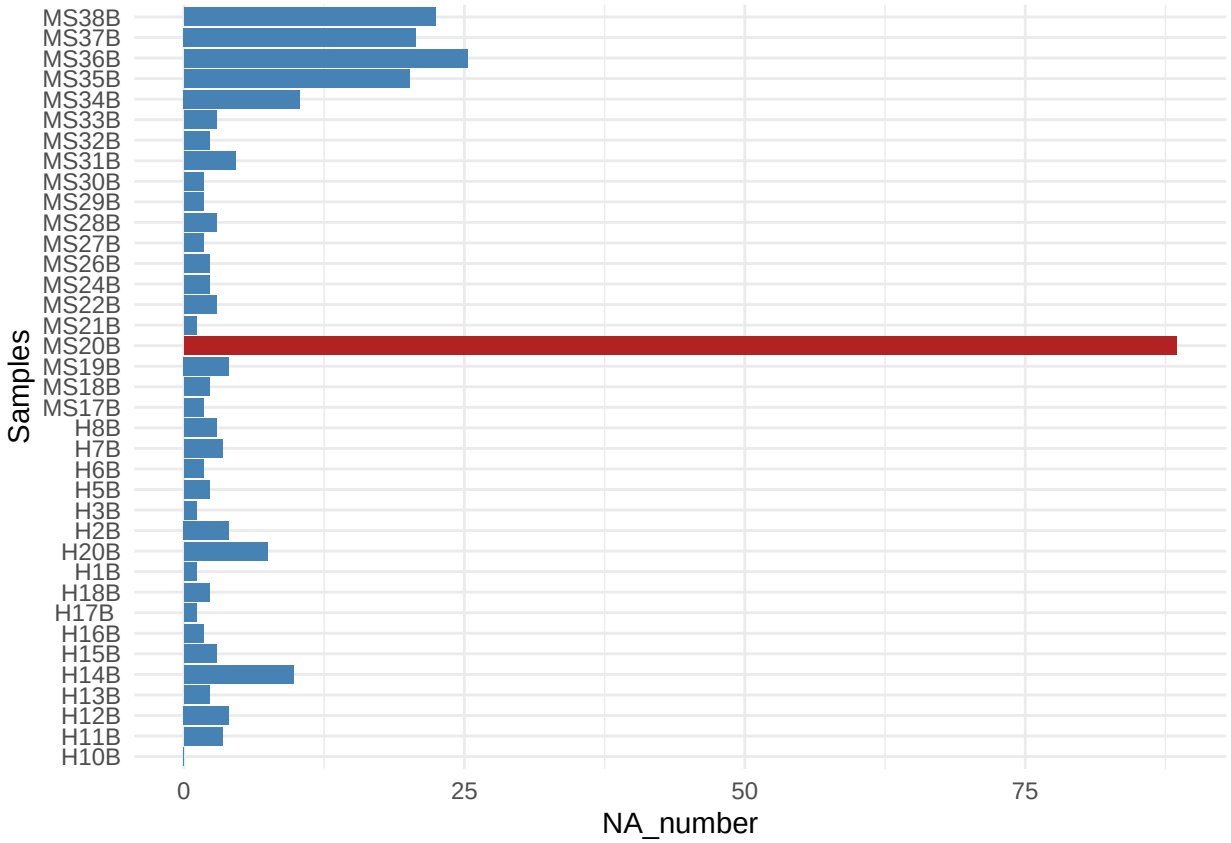
After the tidying up the data, the missing values have to be dealt with. Only proteins that have been detected in at least 70% of the samples will be taken into account.

```
temp <- data.frame(Samples=colnames(data), NA_number=colSums(is.na(data))/nrow(data))
temp=temp[-c(1:3),]
ggplot(temp, aes(x=NA_number, y= Samples)) +
  geom_bar(stat = "identity", fill = ifelse (temp$NA_number >= 0.7 ,
                                             "firebrick","steelblue")) +
  theme_minimal()
```




```
# In red, all the samples with more than 30% of missing values. Compare it afterwards
# the removal of those noisy proteins:

data <- data[which(rowMeans(!is.na(data))> 0.7),]
temp <- data.frame(Samples = colnames(data), NA_number = colSums(is.na(data)) / nrow(data) * 100)
temp=temp[-c(1:3),]
ggplot(temp, aes(x=NA_number, y= Samples)) +
  geom_bar(stat = "identity", fill = ifelse (temp$NA_number >= 70 ,
                                             "firebrick","steelblue")) +
  theme_minimal()
```




```
dim(data)
```

```
## [1] 174  40
```

```
write.table(temp, file = "./data/data_preproccessed.txt", sep = "\t", quote = F, row.names = F)
```

In general, there are not too many missing values after preprocessing,, but 1 sample stands out with a notably higher number than the rest. The MS20 patient has more than 80% of missing values, probably due to experimental errors in the sample. This sample is eliminated for further analysis. After preprocessing, the total number of proteins is 174. Finally, we will reorganize the data in a matrix for the following steps.

```
# Removal of patient MS20
#####
data <- subset(data, select = -MS20B)

# Reorganization of data into a matrix table for next steps
#####
protein.id <- data$Protein.IDs
Gene.names=data$Gene.names
data=data[,-c(1:3)]
protein.id <- gsub(";.+", "",protein.id)  #Keep only the first accession number
rownames(data)=protein.id
data=data.matrix(data)
dim(data)
```

```
## [1] 174  36
```

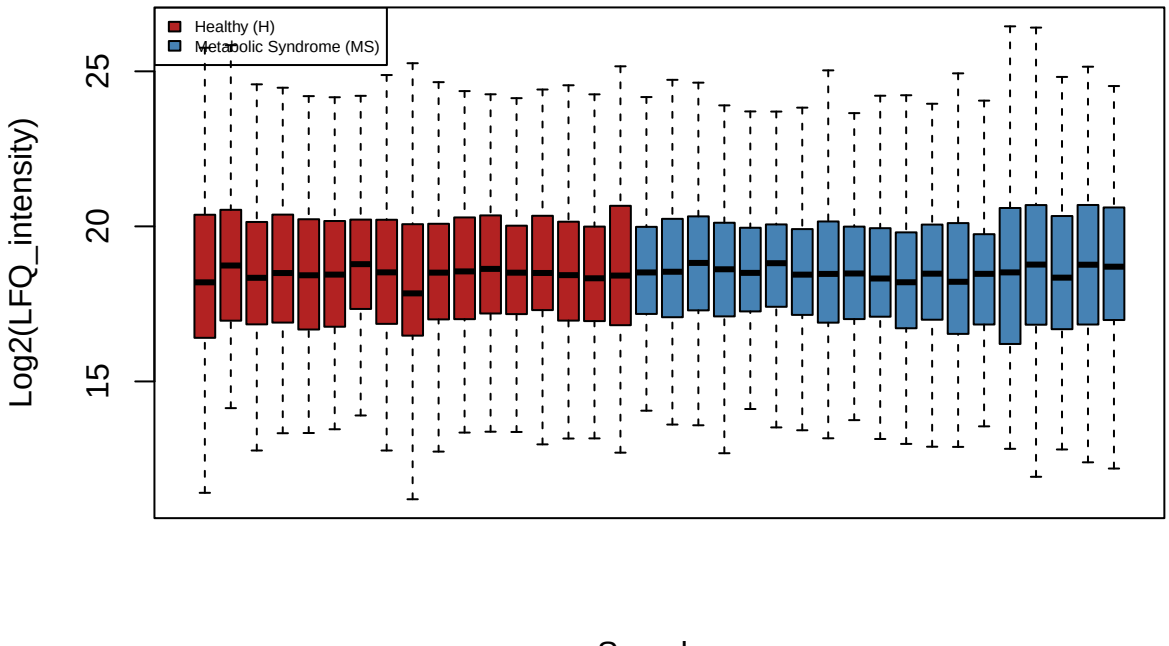
Normalization of the dataset: the distribution of the data should be checked, before and after the normalization. A multiple Shapiro Wilkinson test will be performed to each protein intensity in order to check the normality. The p values obtained will be adjusted by an stringent method (Holm method 1979).

```
counts <- 2^data
p_values <- data.frame(protein.id=rownames(counts),
                       p.value=apply( counts,1, function(x) shapiro.test(x)$p.value))
p_values$adj.p.val <- p.adjust(p_values$p.value, method = "holm")
table(p_values$adj.p.val>0.05)
```

```
##
## FALSE  TRUE
##    56   118
```

We observe that 118 of the 174 proteins conforme to a normal distribution. We visualise the distribution of intensities in each sample before normalisation.

```
y_axis_title = "Log2(LFQ_intensity)"
x_axis_title = "Samples"
main_title = "Boxplot Preprocessed Data"
colors <- c(rep("firebrick", 17), rep("steelblue", 19))
boxplot(data, outline = FALSE, col = colors, cex.lab = 1.5, xaxt = "n")
title(ylab = y_axis_title, xlab=x_axis_title)
title(main = main_title, cex.main = 1.2)
legend("topleft", legend = c("Healthy (H)", "Metabolic Syndrome (MS)"), fill = c("firebrick", "steelblu
```



Since a large number of proteins fit a normal distribution, and the behavior of the samples is quite homogeneous, we decided not to apply any normalization method. We will work with the data transformed to logarithmic scale.

Principal Component Analysis (PCA): Next, in order to identify patterns of interest, detect possible outliers and visualize whether there are different proteomic profiles in both study groups, we performed dimensionality reduction using PCA.

```
# Reorganization of data
#####
pca.data<- data.frame(colnames(data),t(data))
colnames(pca.data)[1] <- "Sample"
pca.data[, -1] <- scale(pca.data[, -1], center = T, scale = F)

pca.data[is.na(pca.data)] <- 0    #Imputation of NA values by zero
table(is.na(pca.data))
```

```
##
## FALSE
##  6300
```

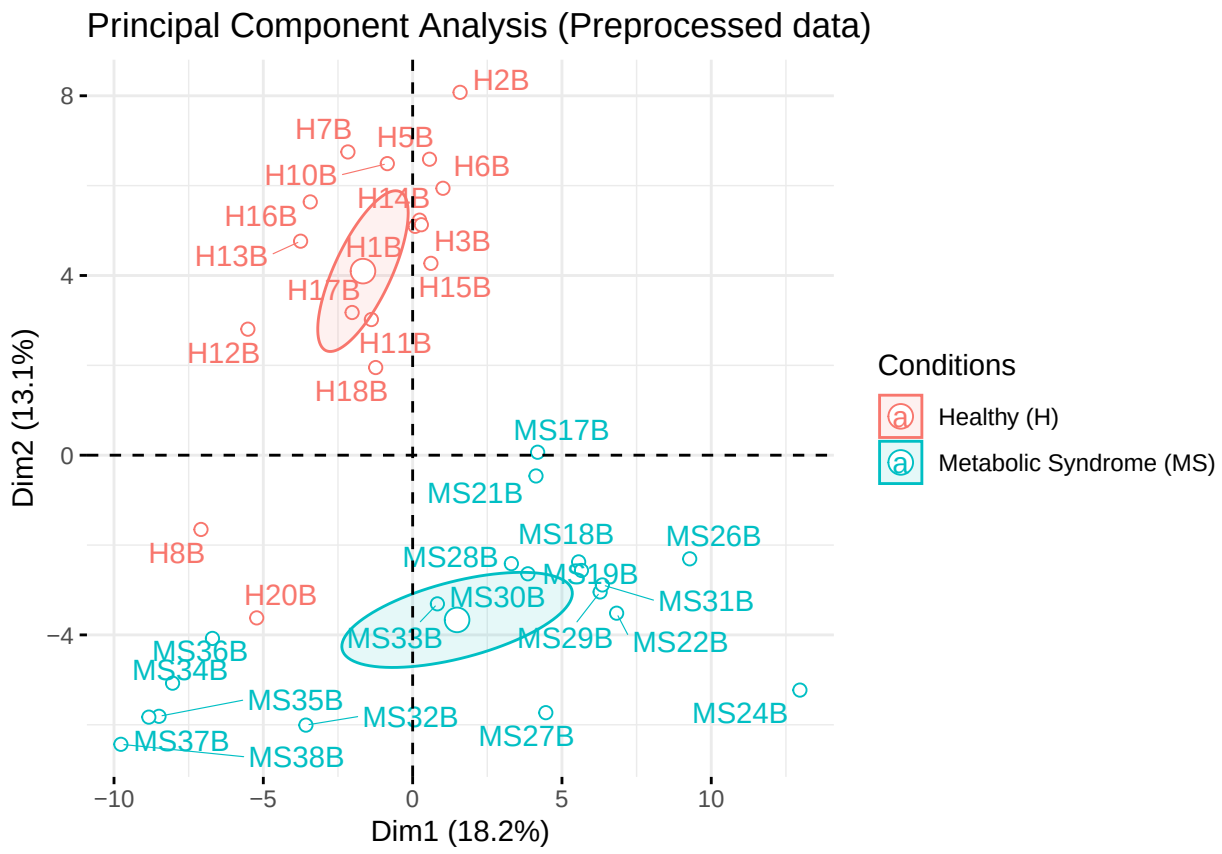
```
pca.data.mixomics=t(pca.data)
data.omics=cbind(Gene.names,pca.data.mixomics[-1,])
write.table(data.omics, file = "./data/preprocessed_data_cero_imputed.txt", sep = "\t", quote = F, row.

# Principal Component Analysis
```

```
#####
res.pca <- PCA(pca.data[,-1], graph = F,scale.unit = F, ncp = 10 )
res.pca$eig
```

```
##          eigenvalue percentage of variance cumulative percentage of variance
## comp 1  28.9899729             18.2113843                           18.21138
## comp 2  20.8329439             13.0871715                           31.29856
## comp 3  13.5106778              8.4873534                           39.78591
## comp 4  12.7586907              8.0149581                           47.80087
## comp 5   9.2032120              5.7814208                           53.58229
## comp 6   7.8136799              4.9085223                           58.49081
## comp 7   6.8601513              4.3095194                           62.80033
## comp 8   5.9268889              3.7232478                           66.52358
## comp 9   4.9974329              3.1393673                           69.66294
## comp 10  4.1908021              2.6326450                           72.29559
## comp 11  4.0677195              2.5553250                           74.85091
## comp 12  3.3456684              2.1017354                           76.95265
## comp 13  3.2885786              2.0658718                           79.01852
## comp 14  3.1066248              1.9515692                           80.97009
## comp 15  2.9672860              1.8640371                           82.83413
## comp 16  2.7904873              1.7529729                           84.58710
## comp 17  2.3549452              1.4793671                           86.06647
## comp 18  2.1997603              1.3818805                           87.44835
## comp 19  2.0251401              1.2721849                           88.72053
## comp 20  1.9566316              1.2291481                           89.94968
## comp 21  1.8685637              1.1738242                           91.12351
## comp 22  1.6258356              1.0213434                           92.14485
## comp 23  1.4720245              0.9247199                           93.06957
## comp 24  1.3838413              0.8693235                           93.93889
## comp 25  1.2763669              0.8018085                           94.74070
## comp 26  1.1993879              0.7534506                           95.49415
## comp 27  1.1751837              0.7382457                           96.23240
## comp 28  0.9949911              0.6250494                           96.85745
## comp 29  0.9289793              0.5835811                           97.44103
## comp 30  0.8754399              0.5499479                           97.99098
## comp 31  0.7988844              0.5018560                           98.49283
## comp 32  0.6857549              0.4307884                           98.92362
## comp 33  0.6511892              0.4090744                           99.33270
## comp 34  0.5463894              0.3432396                           99.67593
## comp 35  0.5158665              0.3240652                          100.00000
```

```
# Individual plot
#####
H <- rep("Healthy (H)", 17)
MS <- rep("Metabolic Syndrome (MS)", 19)
fviz_pca_ind(res.pca, col.ind = c(H,MS),
             pointsize=2, pointshape=21, fill="white",
             repel = TRUE,
             addEllipses = TRUE,ellipse.type = "confidence",
             ellipse.level= 0.95,
             legend.title="Conditions",
             title="Principal Component Analysis (Preprocessed data)",
             show_legend=TRUE,show_guide=TRUE)
```



In order to maintain a conservative approach and avoid risky assumptions, missing values will be imputed by 0. This dataset will be used later for the differential expression analysis and the performance of the sPLS-DA.

```
# Imputation of missing data values to 0
####
data.mean=data
na_values = is.na(data)
data[na_values]=0
```

On the other hand, in order to avoid possible biases in the analysis of linear correlations and logistic regression, we will create a dataset with the missing values imputed by the mean of each group.

```
# Imputation of missing data values by mean
####
h= data.mean[, grepl("^H", colnames(data.mean))]
ms= data.mean[, grepl("MS", colnames(data.mean))]

imp.data.h = h
for (i in 1:nrow(h)){
  imp.data.h[i,is.na(imp.data.h[i,])] <- mean(imp.data.h[i,], na.rm = T)
}

imp.data.ms = ms
for (i in 1:nrow(ms)){
  imp.data.ms[i,is.na(imp.data.ms[i,])] <- mean(imp.data.ms[i,], na.rm = T)
```

```
}

# Download mean imputed dataset for linear correlation and logistic regression analysis
#####
imp.data=cbind(imp.data.h,imp.data.ms)
data.preprocessed=cbind(Gene.names,imp.data)
write.table(data.preprocessed, file = "./data/preprocessed_data_mean_imputed.txt", sep = "\t", quote = F
```

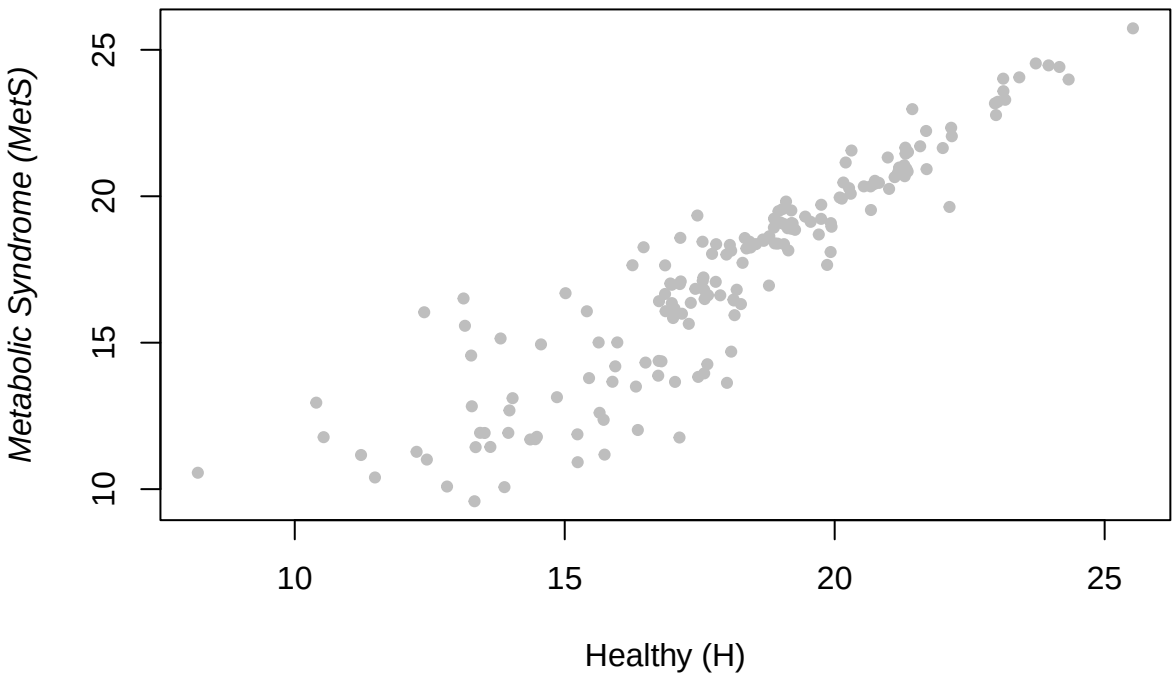
DIFFERENTIAL EXPRESSION ANALYSIS

DIFFERENTIAL EXPRESSION ANALYSIS

Next, in order to detect significant differences in the HDL proteome of MetS subjects, we will perform differential expression analysis by fitting the data to a linear regression model using the **Limma** package.

```
# Elaboration of the average expression matrix
#####
H <- rowMeans(data[, grep("^H", colnames(data))])
MS <- rowMeans(data[, grep("^MS", colnames(data))])
mean.expression <- matrix(c(H,MS),ncol=2)
colnames(mean.expression) <- c("H","MS")
rownames(mean.expression) <- rownames(data)

# Scatterplot to visualize the dispersion between the two conditions
######
plot(H,MS,pch=19,cex=0.7,xlab="Healthy (H)",
     ylab=substitute(italic("Metabolic Syndrome (MetS)")),
     cex.lab=1,
     col="grey")
```



Due to the low number of proteins and the low dispersion observed, we will consider as differentially expressed proteins those with Fold Change $> |0.585|$ and unadjusted p-value < 0.05.

```
# Linear regression model generation
#####
experimental.design <- model.matrix(~ -1 + factor(c(rep(1, 17), rep(2, 19))))
colnames(experimental.design) <- c("H","MS")
linear.fit <- lmFit(data, experimental.design)
contrast.matrix <- makeContrasts(MS-H,levels=c("H","MS"))
contrast.linear.fit <- contrasts.fit(linear.fit, contrast.matrix)
contrast.results <- eBayes(contrast.linear.fit)
nrow(data)
```

```
## [1] 174
```

```
H.MS <- topTable(contrast.results, number=174,coef=1,sort.by="logFC")
log.fold.change <- H.MS$logFC
p.value <- H.MS$P.Value
protein.ids <- rownames(H.MS)
length(protein.ids)
```

```
## [1] 174
```

```r
names(log.fold.change) <- protein.ids
names(p.value) <- protein.ids

# Due to the low scattering seen in the scatterplot, we chose a restrictive value logFC
######
activated.protein <- protein.ids[log.fold.change > 0.585  & p.value < 0.05]
repressed.protein <- protein.ids[log.fold.change < -0.585  & p.value < 0.05]

length(activated.protein)   # 13 diferentially overexpressed proteins
```

```
## [1] 13
```

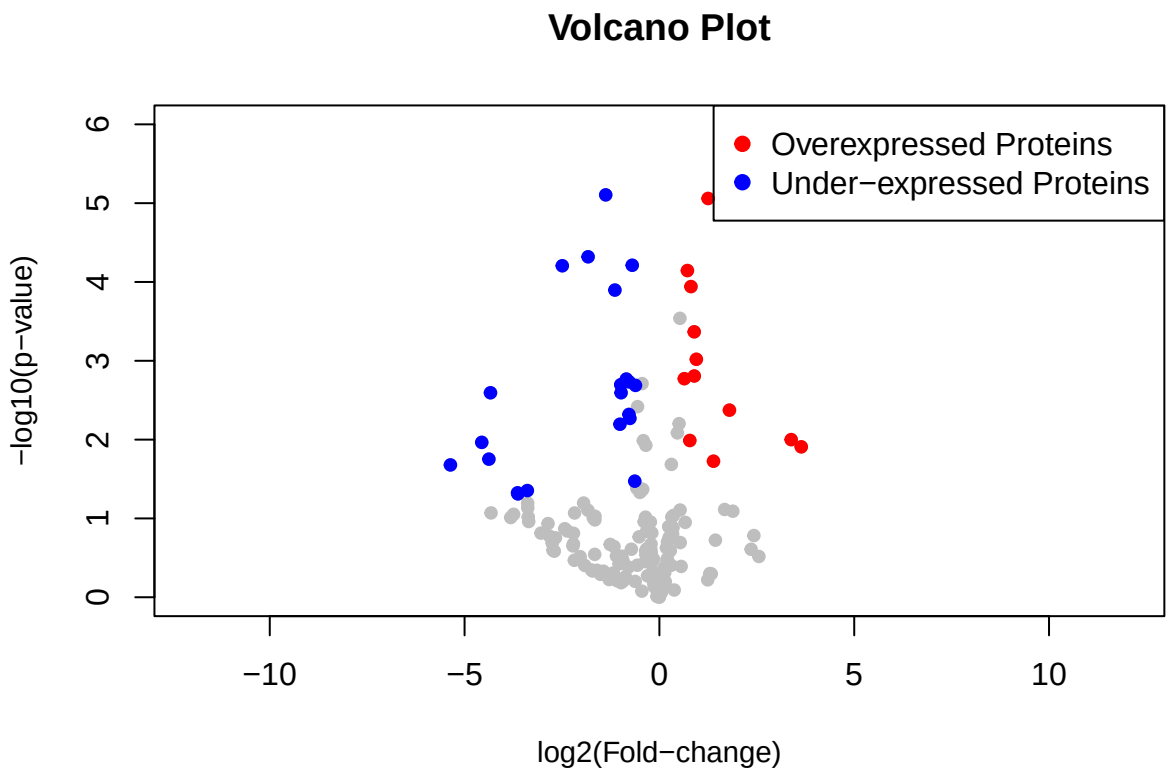
```r
length(repressed.protein)   # 21 diferentially under-expressed proteins
```

```
## [1] 21
```

```r
# Volcano plot visualization
######
log.p.val <- -log10(p.value)
plot(log.fold.change,log.p.val,pch=19,col="grey",cex=0.8,
     xlim=c(-12,12),ylim = c(0,6),
     xlab="log2(Fold-change)",ylab="-log10(p-value)",cex.lab=0.9, main="Volcano Plot")

points(x = log.fold.change[activated.protein],
       y = log.p.val[activated.protein],col="red",cex=0.8,pch=19)
points(x = log.fold.change[repressed.protein],
       y = log.p.val[repressed.protein],col="blue",cex=0.8,pch=19)

legend("topright", legend = c("Overexpressed Proteins", "Under-expressed Proteins"),
       col = c("red", "blue"), pch = 19, cex = 1)
```



13 proteins are overexpressed and 21 proteins are under-expressed in the HDL of MetS subjects.

FUNCTIONAL ENRINCHMENT ANALYSIS

In order to see in which biological processes (BP) the differentially detected proteins are involved, we will perform a functional enrichment analysis using the Gene Ontology (GO) database. Subsequently, we will perform a summary and visualisation of the Biological Process terms using REVIGO.

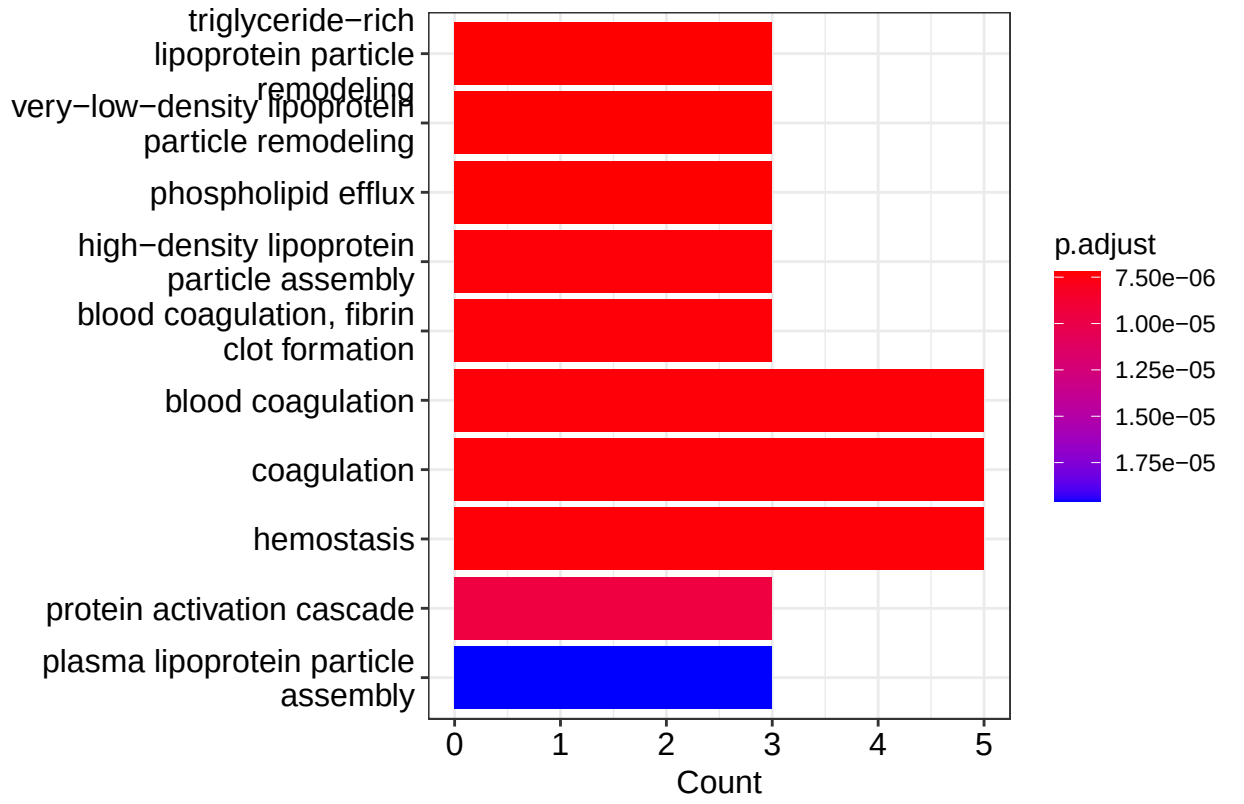
FUNCTIONAL ENRINCHMENT ANALYSIS (OVEREXPRESSED PROTEINS)

```
# We associate each protein with its corresponding gene name
#####
data.analysis=cbind(Gene.names,data)
activated.genes=data.analysis[activated.protein,1]
repressed.genes=data.analysis[repressed.protein,1]


# Enrichment analysis of the differentially activated genes
#####
activated.enrich.go <- enrichGO(gene          = activated.genes,
                                OrgDb         = org.Hs.eg.db,
                                ont           = "BP",
                                pAdjustMethod = "BH",
                                pvalueCutoff  = 0.05,
```

```
                                readable      = FALSE,
                                keyType = "SYMBOL")

# Visualization of enrichment analysys
######
barplot(activated.enrich.go,showCategory = 10,xlab="Number of genes",cex.names=0.2)
```




```
# DOWNLOAD OF THE GO TERMS ASSOCIATED WITH THE ACTIVATED GENES
write.table(x=as.data.frame(activated.enrich.go),
            file="activated.enrich.go.tsv", sep='\t')

# Summary of revigo enrichment
######
revigo.names <- c("term_ID","description","frequency","value","uniqueness","dispensability","representa
revigo.data <- rbind(c("GO:0016126","sterol biosynthetic process",0.24780355936021625,3.950228570029902
                     c("GO:0006633","fatty acid biosynthetic process",0.653300292858752,1.8142253508815
                     c("GO:0006638","neutral lipid metabolic process",0.5406623113313809,3.317391158786
                     c("GO:0006639","acylglycerol metabolic process",0.5293985131786438,3.3173911587861
                     c("GO:0006706","steroid catabolic process",0.14642937598558234,1.3942628166667828,
                     c("GO:0008202","steroid metabolic process",1.4361342644739805,3.545278583711072,0.8
                     c("GO:0016042","lipid catabolic process",1.6445145302996171,2.389846061248035,0.868
                     c("GO:0030258","lipid modification",0.9461590448299165,1.6310854891649278,0.872034
                     c("GO:0046486","glycerolipid metabolic process",1.9767965758053614,2.2231943483879
                     c("GO:0046503","glycerolipid catabolic process",0.2928587519711647,2.4574629192589
                     c("GO:0019068","virion assembly",0.1971164676728993,1.3129157862570044,1,-0,"virio
```

```
c("GO:0031099","regeneration",0.8560486596080198,1.6905424339874104,0.997067735745
c("GO:0007398","ectoderm development",0.10700608245100247,1.4360696455723363,0.997
c("GO:0032102","negative regulation of response to external stimulus",2.331606217
c("GO:0031348","negative regulation of defense response",1.4755575580085605,1.5055
c("GO:0050728","negative regulation of inflammatory response",0.8898400540662311,1
c("GO:0070374","positive regulation of ERK1 and ERK2 cascade",1.2221220995719757,1
c("GO:1902042","negative regulation of extrinsic apoptotic signaling pathway via d
c("GO:1902430","negative regulation of amyloid-beta formation",0.10137418337463391
c("GO:1903034","regulation of response to wounding",0.8954719531425996,1.777112937
c("GO:1903365","regulation of fear response",0.05631899076368552,1.671061244839281
c("GO:1905906","regulation of amyloid fibril formation",0.09574228429826537,1.5411
c("GO:1905907","negative regulation of amyloid fibril formation",0.073214687992791
c("GO:2000352","negative regulation of endothelial cell apoptotic process",0.18022
c("GO:0032374","regulation of cholesterol transport",0.3491777427348502,3.70442185
c("GO:0044058","regulation of digestive system process",0.2421716602838477,2.76673
c("GO:0045807","positive regulation of endocytosis",0.5462942104077495,2.223194348
c("GO:0050708","regulation of protein secretion",1.4586618607794548,2.681241903646
c("GO:1904478","regulation of intestinal absorption",0.06758278891642261,3.5452785
c("GO:1905952","regulation of lipid localization",0.9067357512953367,3.03150321497
c("GO:0033344","cholesterol efflux",0.1520612750619509,3.8490087116510803,0.911745
c("GO:0006898","receptor-mediated endocytosis",0.929263347600811,1.540640489286103
c("GO:0009306","protein secretion",0.6927235863933319,2.331058456250136,0.92945823
c("GO:0010876","lipid localization",2.1513854471727867,3.0799408321203865,0.937549
c("GO:0015748","organophosphate ester transport",0.8222572651498086,3.216635063889
c("GO:0015850","organic hydroxy compound transport",0.9461590448299165,2.537233077
c("GO:0071692","protein localization to extracellular region",0.7377787790042802,2
c("GO:0072578","neurotransmitter-gated ion channel clustering",0.06758278891642261
c("GO:0097062","dendritic spine maintenance",0.06195088984005406,1.451305755006995
c("GO:0034114","regulation of heterotypic cell-cell adhesion",0.1407974769092138,1
c("GO:0034116","positive regulation of heterotypic cell-cell adhesion",0.084478486
c("GO:0034375","high-density lipoprotein particle remodeling",0.09011038522189682,
c("GO:0000302","response to reactive oxygen species",0.9968461365172336,2.916477463
c("GO:0000305","response to oxygen radical",0.10700608245100247,1.3754922896648258
c("GO:0002021","response to dietary excess",0.19148456859653074,1.3235314870752373
c("GO:0002526","acute inflammatory response",0.4336562288803784,2.0972274232298687
c("GO:0006953","acute-phase response",0.22527596305474207,2.700739688145817,0.9445
c("GO:0006979","response to oxidative stress",2.10069835548547,2.13430603954743,0.9
c("GO:0007586","digestion",0.6307726965532777,1.941552076195913,0.9783991089071674
c("GO:0014012","peripheral nervous system axon regeneration",0.03942329353457986,1
c("GO:0030168","platelet activation",0.5631899076368552,3.3173911587861458,0.860813
c("GO:0030195","negative regulation of blood coagulation",0.2590673575129534,2.6841
c("GO:0033194","response to hydroperoxide",0.10137418337463391,1.5226458909353668,
c("GO:0044241","lipid digestion",0.09011038522189682,3.216635063889664,0.955433650
c("GO:0097746","blood vessel diameter maintenance",0.7884658706915972,1.900612165
c("GO:1900272","negative regulation of long-term synaptic potentiation",0.067582788
c("GO:1902950","regulation of dendritic spine maintenance",0.06195088984005406,1.6
c("GO:0035641","locomotory exploration behavior",0.08447848614552828,1.550444999669
c("GO:0001662","behavioral fear response",0.1745888713674251,1.3089214894798682,0.9
c("GO:0002209","behavioral defense response",0.18022077044379364,1.301297631980346
c("GO:0007616","long-term memory",0.2083802658256364,1.3129157862570044,0.946962584
c("GO:0035640","exploration behavior",0.1745888713674251,1.382101674226581,0.964053
c("GO:0042180","cellular ketone metabolic process",0.4787114214913269,1.617564410
c("GO:0042632","cholesterol homeostasis",0.4843433205676954,3.545278583711072,0.97
```

```
                    c("GO:0055088","lipid homeostasis",0.8842081549898626,3.0799408321203865,0.9835026
                    c("GO:0042744","hydrogen peroxide catabolic process",0.14642937598558234,3.0491783
                    c("GO:0032801","receptor catabolic process",0.10137418337463391,1.382101674226581,
                    c("GO:0042159","lipoprotein catabolic process",0.09011038522189682,1.5411275750443
                    c("GO:0046889","positive regulation of lipid biosynthetic process",0.4956071187204
                    c("GO:0010310","regulation of hydrogen peroxide metabolic process",0.1070060824510
                    c("GO:0010896","regulation of triglyceride catabolic process",0.07321468799279117,
                    c("GO:0019216","regulation of lipid metabolic process",1.9035818878125703,2.369482
                    c("GO:0031331","positive regulation of cellular catabolic process",2.3710295111511
                    c("GO:0051043","regulation of membrane protein ectodomain proteolysis",0.135165577
                    c("GO:0051044","positive regulation of membrane protein ectodomain proteolysis",0.
                    c("GO:0060999","positive regulation of dendritic spine development",0.191484568596
                    c("GO:0062012","regulation of small molecule metabolic process",1.819103401667042,
                    c("GO:0062013","positive regulation of small molecule metabolic process",0.8053615
                    c("GO:1902931","negative regulation of alcohol biosynthetic process",0.073214687992
                    c("GO:2000644","regulation of receptor catabolic process",0.06195088984005406,1.635
                    c("GO:0050818","regulation of coagulation",0.4167605316512728,2.3894245296098817,0
                    c("GO:1900221","regulation of amyloid-beta clearance",0.10137418337463391,1.469933
                    c("GO:0051604","protein maturation",1.5938274386123001,3.508717785967395,0.9582273
                    c("GO:0018149","peptide cross-linking",0.1971164676728993,2.9164774634499,0.965787
                    c("GO:0031638","zymogen activation",0.275963054742059,2.461689840892801,0.954305648
                    c("GO:0031639","plasminogen activation",0.05631899076368552,1.384685906130962,0.95
                    c("GO:0051702","biological process involved in interaction with symbiont",0.6364045
                    c("GO:0002227","innate immune response in mucosa",0.1295336787564767,1.382101674226
                    c("GO:0042742","defense response to bacterium",1.6276188330705113,1.32018404344463
                    c("GO:0044403","biological process involved in symbiotic interaction",1.526244649
                    c("GO:0044794","positive regulation by host of viral process",0.11263798152737103,
                    c("GO:0072593","reactive oxygen species metabolic process",0.5913494030186979,1.574
                    c("GO:0097006","regulation of plasma lipoprotein particle levels",0.309754449200270
                    c("GO:0098869","cellular oxidant detoxification",0.4843433205676954,3.545278583711
                    c("GO:0007263","nitric oxide mediated signal transduction",0.11826988060373957,1.3
                    c("GO:0007271","synaptic transmission, cholinergic",0.18585266952016222,1.362986883
                    c("GO:0009636","response to toxic substance",1.2277539986483443,2.697508504013552,
                    c("GO:0019934","cGMP-mediated signaling",0.15769317413831943,1.3129157862570044,0.9
                    c("GO:0032488","Cdc42 protein signal transduction",0.05068709168731696,1.550444999
                    c("GO:0055094","response to lipoprotein particle",0.15769317413831943,1.323531487
                    c("GO:0070371","ERK1 and ERK2 cascade",0.3097544492002703,1.382101674226581,0.9426
                    c("GO:0071402","cellular response to lipoprotein particle stimulus",0.174588871367
                    c("GO:1905918","regulation of CoA-transferase activity",0.04505519261094841,3.7015
                    c("GO:0032770","positive regulation of monooxygenase activity",0.1745888713674251,
                    c("GO:0051006","positive regulation of lipoprotein lipase activity",0.045055192610
                    c("GO:0051341","regulation of oxidoreductase activity",0.5575580085604867,2.134306
                    c("GO:0060191","regulation of lipase activity",0.4956071187204325,2.263752223461731

stuff <- data.frame(revigo.data);
names(stuff) <- revigo.names;

stuff$value <- as.numeric( as.character(stuff$value) );
stuff$frequency <- as.numeric( as.character(stuff$frequency) );
stuff$uniqueness <- as.numeric( as.character(stuff$uniqueness) );
stuff$dispensability <- as.numeric( as.character(stuff$dispensability) );

# Outputs to a PDF file
```

```
pdf( file="revigo_treemap_activated_genes.pdf", width=16, height=9 )

treemap(
  stuff,
  index = c("representative","description"),
  vSize = "value",
  type = "categorical",
  vColor = "representative",
  title = "Revigo TreeMap",
  inflate.labels = FALSE,
  lowerbound.cex.labels = 0,
  bg.labels = 255,
  position.legend = "none"
)

dev.off()
```

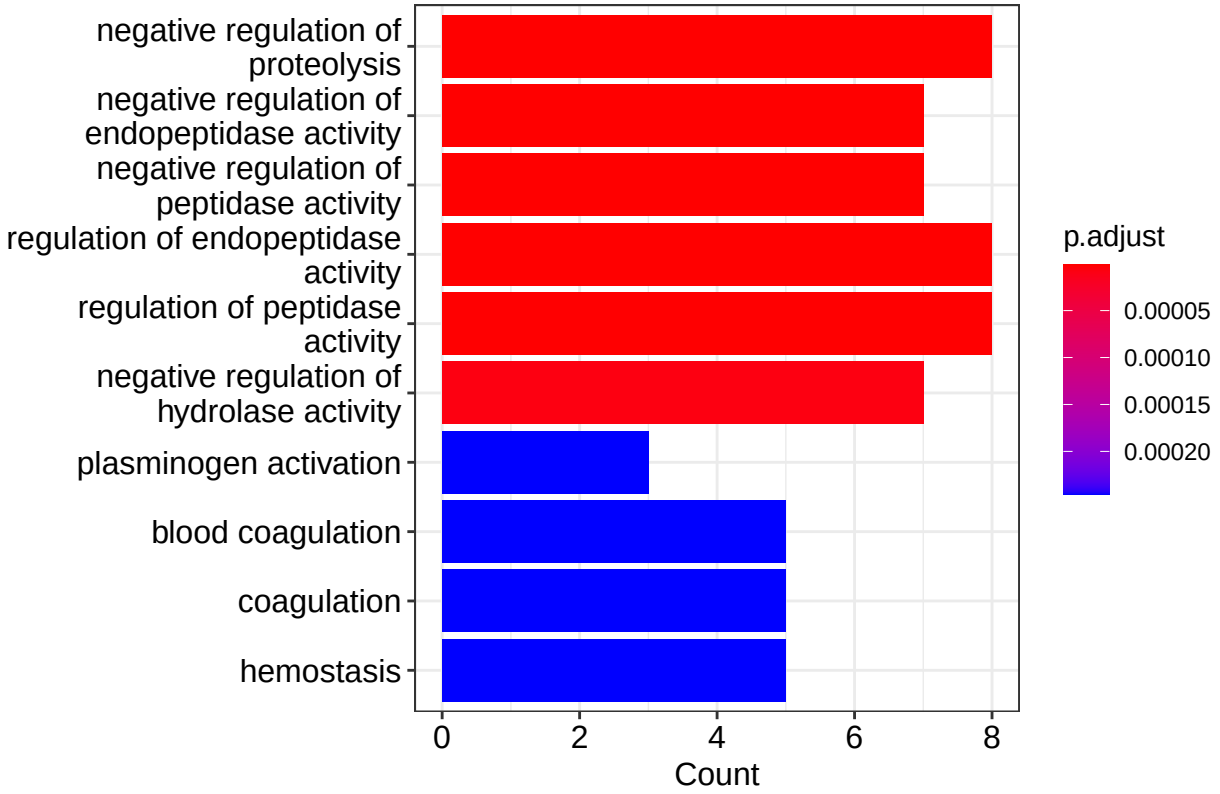
```
## pdf
##   2
```

FUNCTIONAL ENRINCHMENT ANALYSIS (UNDER-EXPRESSED PROTEINS)

```
# Enrichment analysis of the differentially repressed genes
#####
repressed.enrich.go <- enrichGO(gene          = repressed.genes,
                                OrgDb         = org.Hs.eg.db,
                                ont           = "BP",
                                pAdjustMethod = "BH",
                                pvalueCutoff  = 0.05,
                                readable      = FALSE,
                                keyType = "SYMBOL")



# Visualization of enrichment analysys
#####
barplot(repressed.enrich.go,showCategory = 10,xlab="Number of genes")
```




```
# DOWNLOAD OF THE GO TERMS ASSOCIATED WITH THE REPRESSED GENES
write.table(x=as.data.frame(repressed.enrich.go),
            file="repressed.enrich.go.tsv",sep="\t")

# Summary of revigo enrichment
######
revigo.names <- c("term_ID","description","frequency","value","uniqueness","dispensability","representa
revigo.data <- rbind(c("GO:0006956","complement activation",0.27033115566569044,2.0988096479506506,0.771
                     c("GO:0016064","immunoglobulin mediated immune response",0.5012390177968011,1.6287
                     c("GO:0032102","negative regulation of response to external stimulus",2.331606217
                     c("GO:1903034","regulation of response to wounding",0.8954719531425996,1.877031433
                     c("GO:0007596","blood coagulation",0.9855823383644965,3.6097894223878684,0.592323
                     c("GO:0002526","acute inflammatory response",0.4336562288803784,2.212727070759453,
                     c("GO:0006953","acute-phase response",0.22527596305474207,1.7086820433354657,0.917
                     c("GO:0050817","coagulation",0.9968461365172336,3.6097894223878684,0.989161584924
                     c("GO:0050878","regulation of body fluid levels",1.98242847488173,2.660743550620165
                     c("GO:0010755","regulation of plasminogen activation",0.10137418337463391,2.324997
                     c("GO:1903317","regulation of protein maturation",0.3942329353457986,1.418026442131
                     c("GO:0031639","plasminogen activation",0.05631899076368552,3.6097894223878684,0.8
                     c("GO:0006096","glycolytic process",0.24780355936021625,1.3144788491778507,0.895070
                     c("GO:0030212","hyaluronan metabolic process",0.163325073214688,1.9394284997708058
                     c("GO:0031638","zymogen activation",0.275963054742059,2.660743550620165,0.849928279
                     c("GO:0051604","protein maturation",1.5938274386123001,2.0988096479506506,0.920533
                     c("GO:0072376","protein activation cascade",0.10137418337463391,2.249215183825657,
                     c("GO:0034694","response to prostaglandin",0.18022077044379364,2.033808726620647,0
                     c("GO:0035966","response to topologically incorrect protein",0.7940977697679658,1.
```

```
                    c("GO:0050818","regulation of coagulation",0.4167605316512728,2.5931851593715955,0
                    c("GO:0051131","chaperone-mediated protein complex assembly",0.12390177968010813,2

stuff <- data.frame(revigo.data);
names(stuff) <- revigo.names;

stuff$value <- as.numeric( as.character(stuff$value) );
stuff$frequency <- as.numeric( as.character(stuff$frequency) );
stuff$uniqueness <- as.numeric( as.character(stuff$uniqueness) );
stuff$dispensability <- as.numeric( as.character(stuff$dispensability) );

pdf( file="revigo_treemap_repressed_genes.pdf", width=16, height=9 )

treemap(
  stuff,
  index = c("representative","description"),
  vSize = "value",
  type = "categorical",
  vColor = "representative",
  title = "Revigo TreeMap",
  inflate.labels = FALSE,
  lowerbound.cex.labels = 0,
  bg.labels = 255,
  position.legend = "none"
)

dev.off()
```

```
## pdf
##   2
```