aescos_omics_PEC1

2024-11-04

<<<<<<< HEAD
Github repositorio: https://github.com/aescos/Escos-Lopez-Alejandra-PEC1.git

======= >>>>>>> 8d36fb9e244eaba755823a44ce7d45f617bbe1e0
To erase data and start clean:

```
rm(list = ls())
```

To push changes in Git after working on the PEC1.

Los datos que voy a emplear: Descripcion:

"The acompanying dataset has been obtained from a phosphoproteomics experiment that was performed to analyze (3 + 3) PDX models of two different subtypes using Phosphopeptide enriched samples. LC-MS analysis of 2 technical duplicates has been performed on each sample. The results set consisted of Normalized abundances of MS signals for ca. 1400 phosphopeptides Goal of the analysis: *search phosphopeptides that allow differentiation of the two tumor groups This should be made with both Statistical Analysis and visualization. Data have been provided as an excel file: TIO2+PTYR-human-MSS+MSIvsPD.XLSX

Groups are defined as:

MSS group: Samples M1, M5 and T49, PD group: Samples M42, M43 and M64 with two technical replicates for each sample The first column, SequenceModification contains abundance values for the distinct phosphopetides. Other columns can be omitted."

```
library(readxl)
data_phospho <- read_excel("TIO2+PTYR-human-MSS+MSIvsPD.XLSX")
head(data_phospho)
```

```
## # A tibble: 6 × 18
##   SequenceModifications   Accession Description Score M1_1_MSS M1_2_MSS M5_1_MSS
##   <chr>                   <chr>     <chr>       <dbl>    <dbl>    <dbl>    <dbl>
## 1 LYPELSQYMGLSLNEEEIR[2]… O00560    Syntenin-1…  48.1     24.3   44476.        0
## 2 VDKVIQAQTAFSANPANPAILS… O00560    Syntenin-1…  67.0        0   43139.     2102.
## 3 VIQAQTAFSANPANPAILSEAS… O00560    Syntenin-1…  77.7     3413.  172143.   77323.
## 4 HADAEMTGYVVTR[6] Oxida… O15264    Mitogen-ac…  44.9 220431.  145657.  104288.
## 5 HADAEMTGYVVTR[9] Phosp… O15264    Mitogen-ac…  67.4  18255.     8530.   35956.
## 6 STGPGASLGTGYDR[12] Pho… O15551    Claudin-3 …  63.7 644513.  261938.  187023.
## # ℹ 11 more variables: M5_2_MSS <dbl>, T49_1_MSS <dbl>, T49_2_MSS <dbl>,
## #   M42_1_PD <dbl>, M42_2_PD <dbl>, M43_1_PD <dbl>, M43_2_PD <dbl>,
## #   M64_1_PD <dbl>, M64_2_PD <dbl>, CLASS <chr>, PHOSPHO <chr>
```

Las anotaciones sobre los datos son las siguientes:

```
anotacion <- read_excel("TIO2+PTYR-human-MSS+MSIvsPD.XLSX", sheet = "targets")
```

```
## New names:
## • `Sample` -> `Sample...1`
## • `Sample` -> `Sample...2`
```

Voy a testar a ver si hay duplicados primero para poder ver que utilizo como row names.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Duplicados basados en una columna específica
duplicados_peptide_dplyr <- data_phospho %>%
  filter(duplicated(SequenceModifications) | duplicated(SequenceModifications, fromLast = TRUE))

print(duplicados_peptide_dplyr)
```

```
## # A tibble: 2 × 18
##   SequenceModifications   Accession Description Score M1_1_MSS M1_2_MSS M5_1_MSS
##   <chr>                   <chr>     <chr>       <dbl>    <dbl>    <dbl>    <dbl>
## 1 GEPNVSYICSR[7] Phospho… P49840    Glycogen s…  54.3   1.18e7 8689448. 5130833.
## 2 GEPNVSYICSR[7] Phospho… P49840    Glycogen s…  46.9   1.79e4   17796.     3683.
## # ℹ 11 more variables: M5_2_MSS <dbl>, T49_1_MSS <dbl>, T49_2_MSS <dbl>,
## #   M42_1_PD <dbl>, M42_2_PD <dbl>, M43_1_PD <dbl>, M43_2_PD <dbl>,
## #   M64_1_PD <dbl>, M64_2_PD <dbl>, CLASS <chr>, PHOSPHO <chr>
```

Aquí podemos observar que hay un peptido duplicado pero cuando vemos el lugar de fosforilacion en la utlima columna es distinto. El primer lugar de fosforilacion es Y (tirosina) y el segundo es S/T (Serina/Treoina) por lo tanto no esta duplicado el peptido. Por lo tanto voy a unir las dos columnas por un "-" y asi seria una lista unica de peptidos.

```
library(stringr)
data_phospho.unica <- data_phospho %>%
  mutate(unica = str_c(SequenceModifications, PHOSPHO, sep = "-"))
```

Ahora querria tener los nombres de los genes, gene symbol, puesto que es mas sencillo de comprender a la hora de interpretar los resultados. Para ello vamos a utilizar Biomart. Corremos la instalacion del paquete tan solo una vez.

```
library(biomaRt)
```

```
# Conectar al servidor de Ensembl
ensembl <- useMart("ensembl", dataset = "hsapiens_gene_ensembl")
```

```
# Lista de códigos de Accession (uniprotswissprot)
accession_codes <- data_phospho[,2]

# Realizar la consulta
gene_symbols <- getBM(
  attributes = c("ensembl_gene_id","uniprotswissprot", "description","hgnc_symbol","gene_biotype"), # Obtenemos d
istintos codigos para el mismo gen.
  filters = "uniprotswissprot",
  values = accession_codes,
  mart = ensembl
)

# Mostrar resultados
head(gene_symbols)
```

```
##   ensembl_gene_id uniprotswissprot
## 1 ENSG00000156603           A0JLT2
## 2 ENSG00000188522           A6ND36
## 3 ENSG00000196531           E9PAV3
## 4 ENSG00000110696           O00193
## 5 ENSG00000101856           O00264
## 6 ENSG00000103319           O00418
##                                                                        description
## 1                        mediator complex subunit 19 [Source:HGNC Symbol;Acc:HGNC:29600]
## 2         family with sequence similarity 83 member G [Source:HGNC Symbol;Acc:HGNC:32554]
## 3 nascent polypeptide associated complex subunit alpha [Source:HGNC Symbol;Acc:HGNC:7629]
## 4                  chromosome 11 open reading frame 58 [Source:HGNC Symbol;Acc:HGNC:16990]
## 5           progesterone receptor membrane component 1 [Source:HGNC Symbol;Acc:HGNC:16090]
## 6              eukaryotic elongation factor 2 kinase [Source:HGNC Symbol;Acc:HGNC:24615]
##   hgnc_symbol   gene_biotype
## 1       MED19 protein_coding
## 2      FAM83G protein_coding
## 3        NACA protein_coding
## 4    C11orf58 protein_coding
## 5      PGRMC1 protein_coding
## 6       EEF2K protein_coding
```

Unimos ambas listas para que contenga toda la informacion para cada peptido.

```
# Cambiar el nombre de la columna con dplyr
gene_symbols <- dplyr::rename(gene_symbols, Accession = uniprotswissprot)
```

```
gene_symbols <- gene_symbols %>% distinct(Accession, .keep_all = TRUE)

# Union izquierda, en este caso por data_phospho.unica
data <- inner_join(data_phospho.unica, gene_symbols, by = "Accession")

# Eliminar duplicados basados en la columna "unica"
data <- data %>% distinct(unica, .keep_all = TRUE)
print(data)
```

```
## # A tibble: 1,437 × 23
##    SequenceModifications  Accession Description Score M1_1_MSS M1_2_MSS M5_1_MSS
##    <chr>                  <chr>     <chr>       <dbl>    <dbl>    <dbl>    <dbl>
##  1 LYPELSQYMGLSLNEEEIR[2… O00560    Syntenin-1…  48.1     24.3   44476.        0
##  2 VDKVIQAQTAFSANPANPAIL… O00560    Syntenin-1…  67.0        0   43139.     2102.
##  3 VIQAQTAFSANPANPAILSEA… O00560    Syntenin-1…  77.7     3413.  172143.   77323.
##  4 HADAEMTGYVVTR[6] Oxid… O15264    Mitogen-ac…  44.9 220431.  145657.  104288.
##  5 HADAEMTGYVVTR[9] Phos… O15264    Mitogen-ac…  67.4  18255.     8530.   35956.
##  6 STGPGASLGTGYDR[12] Ph… O15551    Claudin-3 …  63.7 644513.  261938.  187023.
##  7 DHVYGIHNPVMTSPSQH[4] … O43490    Prominin-1…  40.7 686820.  331984.  252694.
##  8 DHVYGIHNPVMTSPSQH[4] … O43490    Prominin-1…  58.3 815186.  728701.  267179.
##  9 RMDSEDVYDDVETIPMK[2] … O43490    Prominin-1…  60.9    1578.     9836.    2033.
## 10 RMDSEDVYDDVETIPMK[2] … O43490    Prominin-1…  47.4    2815.    13247.    2121.
## # ℹ 1,427 more rows
## # ℹ 16 more variables: M5_2_MSS <dbl>, T49_1_MSS <dbl>, T49_2_MSS <dbl>,
## #   M42_1_PD <dbl>, M42_2_PD <dbl>, M43_1_PD <dbl>, M43_2_PD <dbl>,
## #   M64_1_PD <dbl>, M64_2_PD <dbl>, CLASS <chr>, PHOSPHO <chr>, unica <chr>,
## #   ensembl_gene_id <chr>, description <chr>, hgnc_symbol <chr>,
## #   gene_biotype <chr>
```

Poner los datos en formato SummarizedExperiment:

Instalamos e inicializamos la libreria.

```
# Seleccionar columnas de la matriz
data <-data.frame(data, row.names = "unica")
matrix <- dplyr::select(data, M1_1_MSS, M1_2_MSS, M5_1_MSS, M5_2_MSS, T49_1_MSS, T49_2_MSS, M42_1_PD, M42_2_PD, M
43_1_PD, M43_2_PD, M64_1_PD, M64_2_PD)
matrix <- matrix %>% as.matrix()

# Metadatos de las filas (genes)
data <-data.frame(data)
row_data <- data.frame(
  ensembl_gene_id = data[,19],
  description = data[,20],
  row.names = row.names(data),
  Symbol = data[,21]
)

# Metadatos de las columnas (muestras)
col_data <- anotacion
```

```
# Crear el objeto SummarizedExperiment
se <- SummarizedExperiment(
  assays = list(counts = matrix),  # Asignar los datos de expresión
  rowData = row_data,              # Asignar los metadatos de los genes
  colData = col_data               # Asignar los metadatos de las muestras
)

# Mostrar el objeto
print(se)
```

```
## class: SummarizedExperiment
## dim: 1437 12
## metadata(0):
## assays(1): counts
## rownames(1437): LYPELSQYMGLSLNEEEIR[2] Phospho|[9] Oxidation-Y
##   VDKVIQAQTAFSANPANPAILSEASAPIPHDGNLYPR[35] Phospho-Y ...
##   YQDEVFGGFVTEPQEESEEEVEEPEER[17] Phospho-S/T YSPSQNSPIHHIPSRR[1]
##   Phospho|[7] Phospho-S/T
## rowData names(3): ensembl_gene_id description Symbol
## colnames(12): M1_1_MSS M1_2_MSS ... M64_1_PD M64_2_PD
## colData names(4): Sample...1 Sample...2 Individual Phenotype
```

```
save(se, file = "Phosphoproteomics.Rda")
```

Estadisticos descriptivos

```
library(knitr)
library(kableExtra)
```

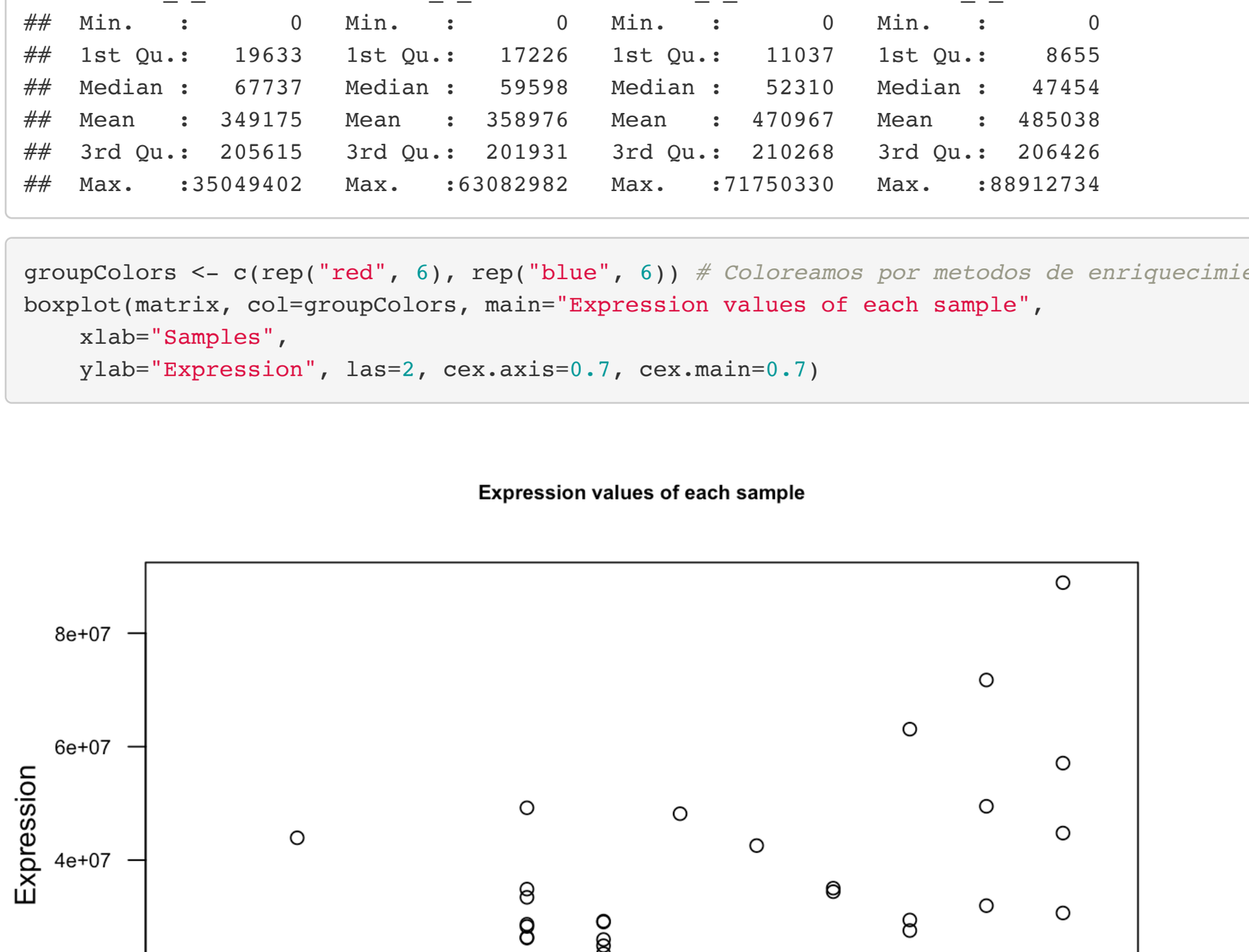
```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
summary(matrix)
```

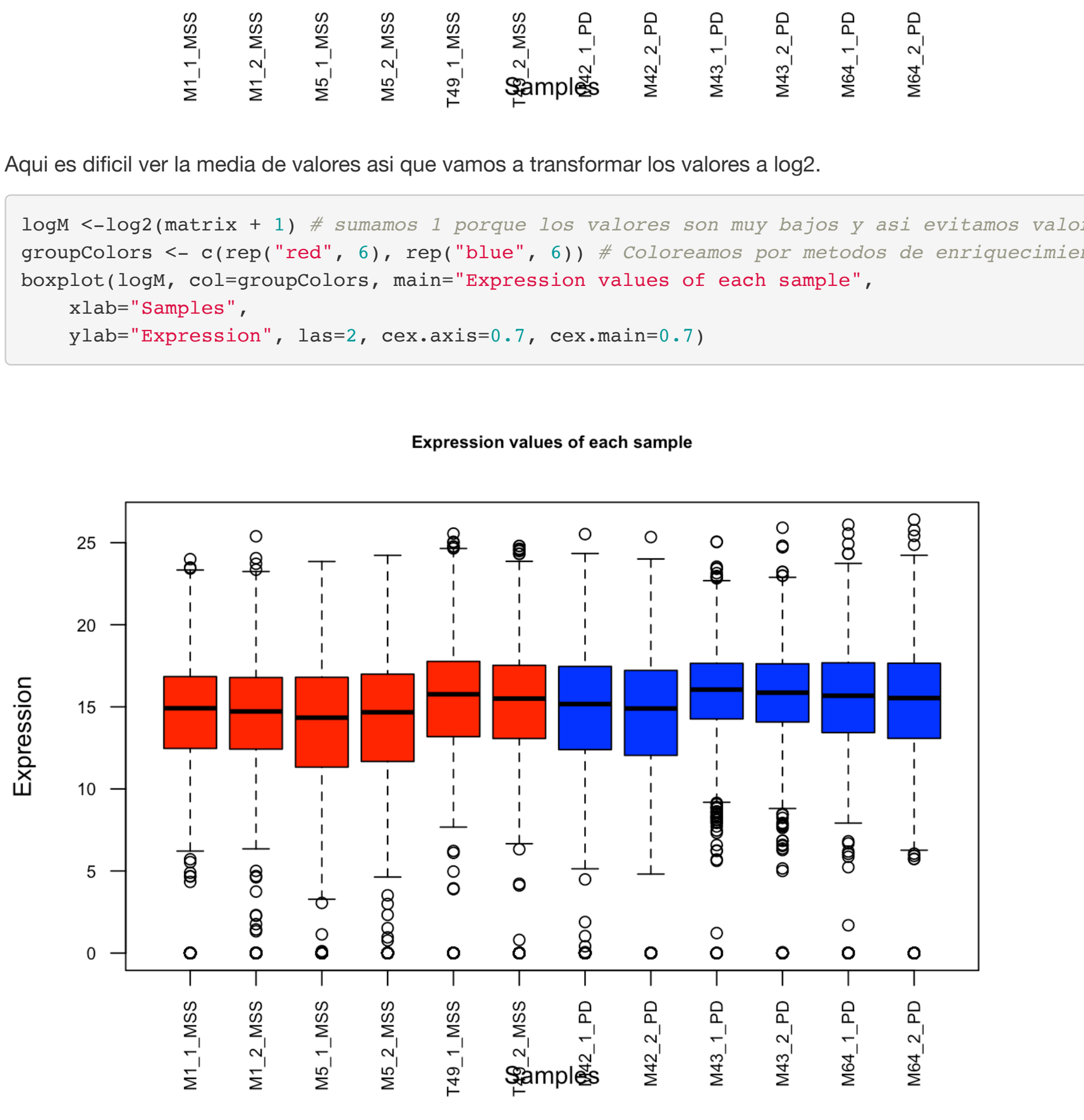
```
##     M1_1_MSS          M1_2_MSS          M5_1_MSS          M5_2_MSS
##  Min.   :       0   Min.   :       0   Min.   :       0   Min.   :       0
##  1st Qu.:    5651   1st Qu.:    5488   1st Qu.:    2567   1st Qu.:    3261
##  Median :   30871   Median :   27001   Median :   20749   Median :   26067
##  Mean   :  229986   Mean   :  253312   Mean   :  233072   Mean   :  261212
##  3rd Qu.:  117475   3rd Qu.:  113194   3rd Qu.:  114138   3rd Qu.:  130208
##  Max.   :16719906   Max.   :43928481   Max.   :15135169   Max.   :19631820
##    T49_1_MSS         T49_2_MSS          M42_1_PD          M42_2_PD
##  Min.   :       0   Min.   :       0   Min.   :       0   Min.   :       0
##  1st Qu.:    9293   1st Qu.:    8607   1st Qu.:    5393   1st Qu.:    4214
##  Median :   55654   Median :   46397   Median :   36887   Median :   30597
##  Mean   :  542800   Mean   :  462909   Mean   :  388693   Mean   :  333813
##  3rd Qu.:  223267   3rd Qu.:  189197   3rd Qu.:  180508   3rd Qu.:  152696
##  Max.   :49218872   Max.   :29240206   Max.   :48177680   Max.   :42558111
##     M43_1_PD          M43_2_PD          M64_1_PD          M64_2_PD
##  Min.   :       0   Min.   :       0   Min.   :       0   Min.   :       0
##  1st Qu.:   19633   1st Qu.:   17226   1st Qu.:   11037   1st Qu.:    8655
##  Median :   67737   Median :   59598   Median :   52310   Median :   47454
##  Mean   :  349175   Mean   :  358976   Mean   :  470967   Mean   :  485038
##  3rd Qu.:  205615   3rd Qu.:  201931   3rd Qu.:  210268   3rd Qu.:  206426
##  Max.   :35049402   Max.   :63082982   Max.   :71750330   Max.   :88912734
```

```
groupColors <- c(rep("red", 6), rep("blue", 6)) # Coloreamos por metodos de enriquecimiento de phosphopeptidos.
boxplot(matrix, col=groupColors, main="Expression values of each sample",
    xlab="Samples",
    ylab="Expression", las=2, cex.axis=0.7, cex.main=0.7)
```



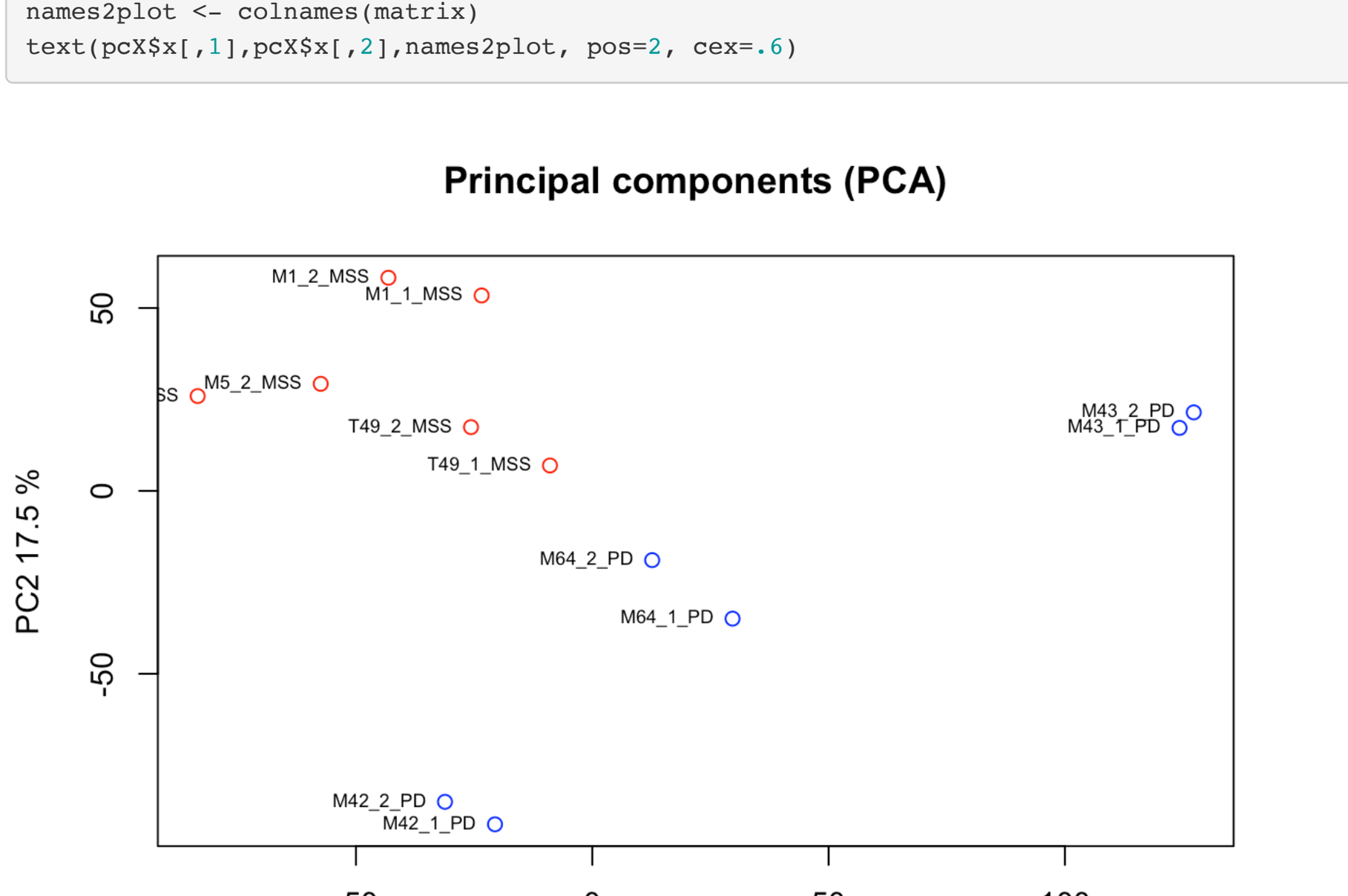
Aqui es dificil ver la media de valores asi que vamos a transformar los valores a log2.

```
logM <-log2(matrix + 1) # sumamos 1 porque los valores son muy bajos y asi evitamos valores proximos a 0.
groupColors <- c(rep("red", 6), rep("blue", 6)) # Coloreamos por metodos de enriquecimiento de phosphopeptidos.
boxplot(logM, col=groupColors, main="Expression values of each sample",
    xlab="Samples",
    ylab="Expression", las=2, cex.axis=0.7, cex.main=0.7)
```



Aqui podemos ver que los duplicados se parecen entre ellos lo cual es lo ideal.

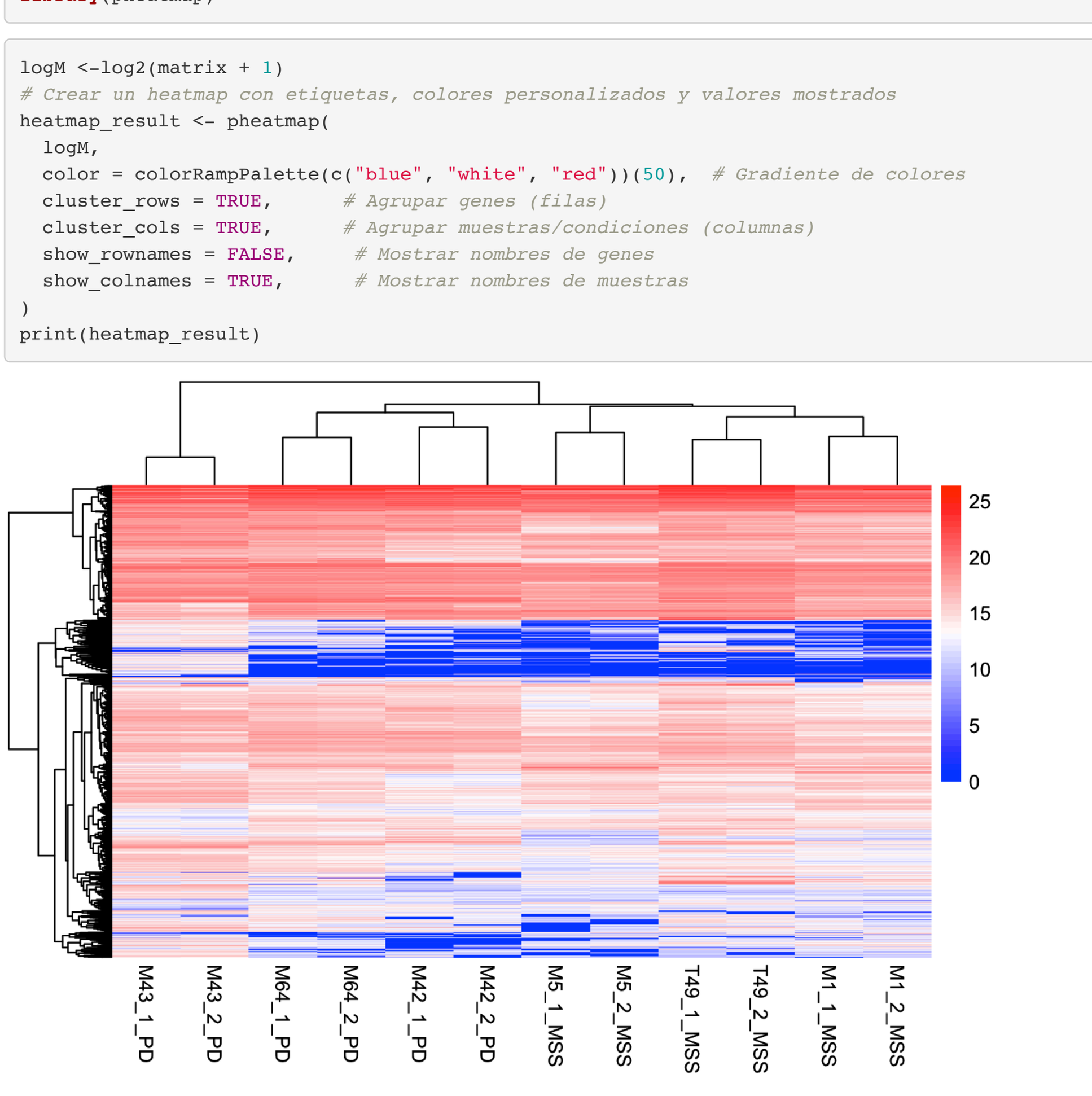
```
pcX<-prcomp(t(logM), scale=FALSE) # Ya se han escalado los datos
loads<- round(pcX$sdev^2/sum(pcX$sdev^2)*100,1)
# Then plot the first two components.

xlab<-c(paste("PC1",loads[1],"%"))
ylab<-c(paste("PC2",loads[2],"%"))
plot(pcX$x[,1:2],xlab=xlab,ylab=ylab, col=groupColors,
     main ="Principal components (PCA)")
#names2plot<-paste0(substr(names(matrix),1,3), 1:6)
names2plot <- colnames(matrix)
text(pcX$x[,1],pcX$x[,2],names2plot, pos=2, cex=.6)
```



Aqui podemos observar que las muestras duplicadas no se replican muy bien y que que como mucho se separan las muestras por PD y MSS que son dos metodos distintos de enriquecer los fosfopeptidos.

```
library(pheatmap)
```

```
logM <-log2(matrix + 1)
# Crear un heatmap con etiquetas, colores personalizados y valores mostrados
heatmap_result <- pheatmap(
  logM,
  color = colorRampPalette(c("blue", "white", "red"))(50),  # Gradiente de colores
  cluster_rows = TRUE,       # Agrupar genes (filas)
  cluster_cols = TRUE,       # Agrupar muestras/condiciones (columnas)
  show_rownames = FALSE,      # Mostrar nombres de genes
  show_colnames = TRUE,       # Mostrar nombres de muestras
)
print(heatmap_result)
```



Podemos observar que se agrupan el metodo de enriquecimiento de peptidos y por duplicados de muestra. Los heatmaps son muy utiles para analizar patrones mas profundamente.