Análisis de Expresión Diferencial de los Hepatocitos con respecto a la Edad

José María de Madaria López

20/06/2023

Contents

Librerias

```
library(anndata)
library(Seurat)
```

```
## Attaching SeuratObject
```

```
library(edgeR)
```

```
## Loading required package: limma
```

```
library(DESeq2)
```

```
## Loading required package: S4Vectors
```

```
## 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: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
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:utils':
##
##     findMatches
```

```
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
```

```
## Loading required package: IRanges
```

```
##
## Attaching package: 'IRanges'
```

```
## The following object is masked from 'package:grDevices':
##
##     windows
```

```
## Loading required package: GenomicRanges
```

```
## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## 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")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians

##
## Attaching package: 'SummarizedExperiment'
```

```
## The following object is masked from 'package:SeuratObject':
##
##     Assays

## The following object is masked from 'package:Seurat':
##
##     Assays
```

```
library(ggplot2)
library(gplots)
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:IRanges':
##
##     space

## The following object is masked from 'package:S4Vectors':
##
##     space

## The following object is masked from 'package:stats':
##
##     lowess
```

Creación del objeto Seurat

```
# Leemos el archivo .h5ad que proviene de python.
setwd("C:/Users/pepi/Desktop/Python-TFM")
data_age <- read_h5ad("age_adata.h5ad")
seurat_age <- CreateSeuratObject(counts = t(data_age$X),
                                 meta.data = data_age$obs)

print(str(seurat_age))
```

```
## Formal class 'Seurat' [package "SeuratObject"] with 13 slots
##   ..@ assays      :List of 1
##   .. ..$ RNA:Formal class 'Assay' [package "SeuratObject"] with 8 slots
##   .. .. .. .. ..@ counts       :Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
##   .. .. .. .. .. .. ..@ i       : int [1:38718] 0 1 2 3 4 5 9 10 11 12 ...
##   .. .. .. .. .. .. ..@ p       : int [1:7] 0 7703 13859 20659 25983 32277 38718
##   .. .. .. .. .. .. ..@ Dim     : int [1:2] 9750 6
##   .. .. .. .. .. .. ..@ Dimnames:List of 2
##   .. .. .. .. .. .. .. ..$ : chr [1:9750] "ENSG00000000003" "ENSG00000000419" "ENSG00000000938" "ENSG000
##   .. .. .. .. .. .. .. ..$ : chr [1:6] "hepatocyte_40 year" "hepatocyte_46 year" "hepatocyte_58 year" "
##   .. .. .. .. .. .. ..@ x       : num [1:38718] 175.4 135.7 8.1 386.1 55.9 ...
##   .. .. .. .. .. .. ..@ factors : list()
##   .. .. .. ..@ data         :Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
```

```
##   .. .. .. .. .. ..@ i       : int [1:38718] 0 1 2 3 4 5 9 10 11 12 ...
##   .. .. .. .. .. ..@ p       : int [1:7] 0 7703 13859 20659 25983 32277 38718
##   .. .. .. .. .. ..@ Dim     : int [1:2] 9750 6
##   .. .. .. .. .. ..@ Dimnames:List of 2
##   .. .. .. .. .. .. ..$ : chr [1:9750] "ENSG00000000003" "ENSG00000000419" "ENSG00000000938" "ENSG000
##   .. .. .. .. .. .. ..$ : chr [1:6] "hepatocyte_40 year" "hepatocyte_46 year" "hepatocyte_58 year" "
##   .. .. .. .. .. ..@ x       : num [1:38718] 175.4 135.7 8.1 386.1 55.9 ...
##   .. .. .. .. .. ..@ factors : list()
##   .. .. .. ..@ scale.data   : num[0 , 0 ]
##   .. .. .. ..@ key          : chr "rna_"
##   .. .. .. ..@ assay.orig   : NULL
##   .. .. .. ..@ var.features : logi(0)
##   .. .. .. ..@ meta.features:'data.frame':   9750 obs. of  0 variables
##   .. .. .. ..@ misc         : list()
##   ..@ meta.data   :'data.frame': 6 obs. of  7 variables:
##   .. ..$ orig.ident  : Factor w/ 1 level "hepatocyte": 1 1 1 1 1 1
##   .. ..$ nCount_RNA  : num [1:6] 1050074 385320 588594 149169 285907 ...
##   .. ..$ nFeature_RNA: int [1:6] 7703 6156 6800 5324 6294 6441
##   .. ..$ sex         : Factor w/ 2 levels "female","male": 2 1 2 2 1 2
##   .. ..$ individual  : chr [1:6] "donor2" "donor6" "donor4" "donor3" ...
##   .. ..$ cell_type   : Factor w/ 1 level "hepatocyte": 1 1 1 1 1 1
##   .. ..$ age         : chr [1:6] "40 year" "46 year" "58 year" "66 year" ...
##   ..@ active.assay: chr "RNA"
##   ..@ active.ident: Factor w/ 1 level "hepatocyte": 1 1 1 1 1 1
##   .. ..- attr(*, "names")= chr [1:6] "hepatocyte_40 year" "hepatocyte_46 year" "hepatocyte_58 year" "
##   ..@ graphs      : list()
##   ..@ neighbors   : list()
##   ..@ reductions  : list()
##   ..@ images      : list()
##   ..@ project.name: chr "SeuratProject"
##   ..@ misc        : list()
##   ..@ version     :Classes 'package_version', 'numeric_version'  hidden list of 1
##   .. ..$ : int [1:3] 4 1 3
##   ..@ commands    : list()
##   ..@ tools       : list()
## NULL
```

```
counts_age <- seurat_age@assays$RNA@counts
x<- data.frame(counts_age)

str(x)
```

```
## 'data.frame':    9750 obs. of  6 variables:
##  $ hepatocyte_40.year: num  175.4 135.7 8.1 386.1 55.9 ...
##  $ hepatocyte_46.year: num  69.17 58.65 2.62 282.55 14.97 ...
##  $ hepatocyte_58.year: num  134.7 78.4 12.1 336.3 33.5 ...
##  $ hepatocyte_66.year: num  0 20.6 11.1 100.7 0 ...
##  $ hepatocyte_69.year: num  62.86 30.16 6.17 175.32 15.31 ...
##  $ hepatocyte_84.year: num  129.99 104.23 8.32 333.08 47.12 ...
```

```
dim(x)
```

```
## [1] 9750    6
```

```
head(x)
```

```
##                 hepatocyte_40.year hepatocyte_46.year hepatocyte_58.year
## ENSG00000000003         175.390259          69.165359          134.70703
## ENSG00000000419         135.673859          58.649395           78.40900
## ENSG00000000938           8.096465           2.618579           12.07803
## ENSG00000000971         386.126892         282.549835          336.31369
## ENSG00000001036          55.859879          14.967182           33.49059
## ENSG00000001084          86.122429          50.436680           72.20681
##                 hepatocyte_66.year hepatocyte_69.year hepatocyte_84.year
## ENSG00000000003            0.00000          62.856647          129.98643
## ENSG00000000419           20.56833          30.159922          104.23354
## ENSG00000000938           11.09921           6.168148            8.31739
## ENSG00000000971          100.69341         175.318497          333.08319
## ENSG00000001036            0.00000          15.309212           47.11673
## ENSG00000001084            0.00000          49.949757           50.77285
```

Filtrado

```
# Filtramos eliminando los counts iguales a 0.
suma_rows_x <- rowSums(x)
nueva_matriz <- x[suma_rows_x != 0,]
dim(nueva_matriz)
```

```
## [1] 9396    6
```

Creación del objeto DESeqDataSet

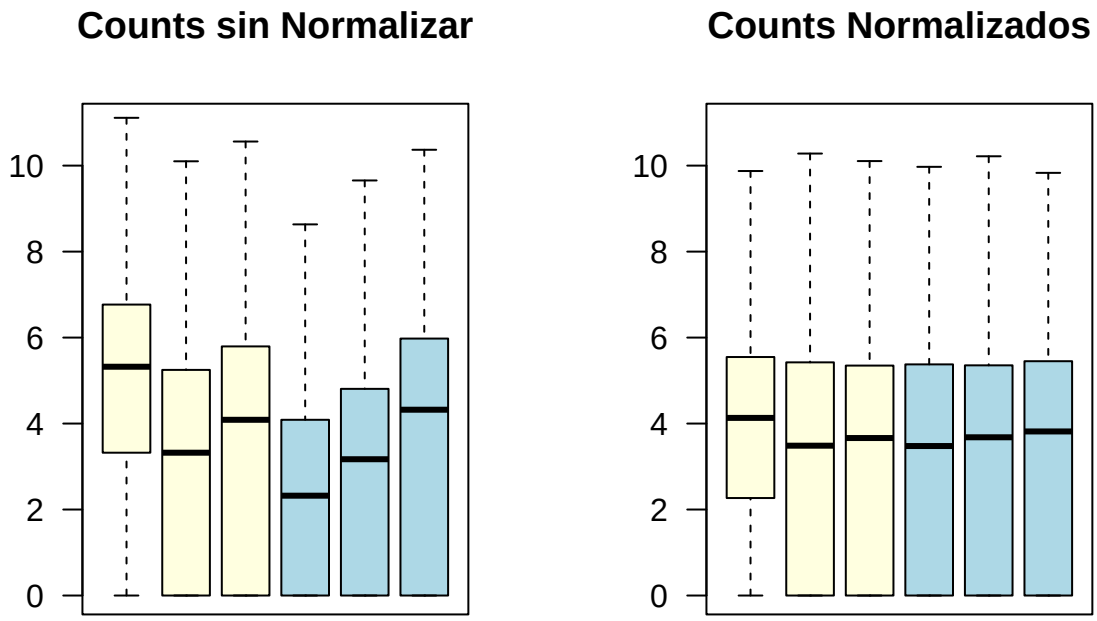
```
# Creamos el coldata necesario para realizar matriz DESeqData
age <- c(40, 46, 58, 66, 69, 84)

grupos <-c("H40_50","H40_50","H40_50","H60_80",
           "H60_80","H60_80")
colData <- data.frame(grupos, age)
DESeqData <- DESeqDataSetFromMatrix(countData = round(nueva_matriz),
                                    colData = colData,
                                    design = ~ grupos)
```

```
## converting counts to integer mode
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
# De modo visual podemos apreciar los datos normalizados mediante un boxplot:
par(mfrow=c(1,2))
boxplot(log2(counts(DESeqData,normalized=FALSE)+1),
```

```
        main="Counts sin Normalizar",xaxt = "n",
        col=rep(c("lightyellow","lightblue"),
        each=3), las=2, ylim = c(0, 11))

DESeqData_2 <- estimateSizeFactors(DESeqData)
boxplot(log2(counts(DESeqData_2,normalized=TRUE)+1),xaxt = "n",
        main="Counts Normalizados",
        col=rep(c("lightyellow","lightblue"),
                each=3),las=2, ylim = c(0, 11))
```




```
par(mfrow=c(1,1))

DESeqData <- estimateSizeFactors(DESeqData)
counts_normalizados <- counts(DESeqData, normalized = TRUE)


DESeqData <- DESeqDataSetFromMatrix(countData = round(counts_normalizados),
                                    colData = colData,
                                    design = ~ grupos)
```

```
## converting counts to integer mode

## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

Pruebas DESeq2

Análsis exploratorio

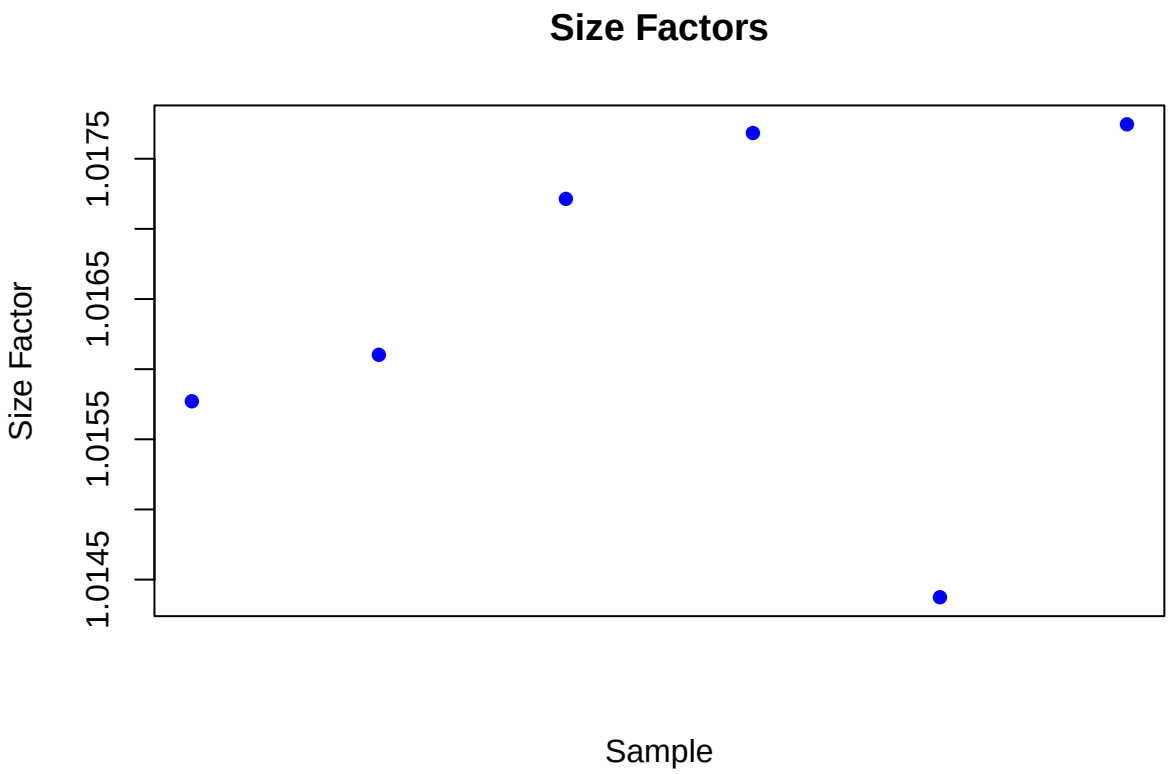
```
# Creación de la matriz DESeq2.
# Realizar el análisis de expresión diferencial
DESeqData <- DESeq(DESeqData)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## -- note: fitType='parametric', but the dispersion trend was not well captured by the
##    function: y = a/x + b, and a local regression fit was automatically substituted.
##    specify fitType='local' or 'mean' to avoid this message next time.

## final dispersion estimates

## fitting model and testing
```

```
# Extraer los size factors del objeto DESeqDataSet y asignar nombres a
# las muestras
size<-sizeFactors(DESeqData)
size
```

```
## hepatocyte_40.year hepatocyte_46.year hepatocyte_58.year hepatocyte_66.year
##           1.015771           1.016102           1.017214           1.017683
## hepatocyte_69.year hepatocyte_84.year
##           1.014374           1.017745
```

```
names(sizeFactors) <- colnames(counts(DESeqData))

# Gráfico que muestra la profundidad de secuenciacion de cada muestra
plot(size, pch = 16, col = "blue", main = "Size Factors",
     xlab = "Sample", ylab = "Size Factor", xaxt = "n")
```




```
# Averiguamos los resultados:
Resultados_Age <- results(DESeqData)
head(Resultados_Age)
```

```
## log2 fold change (MLE): grupos H60 80 vs H40 50
## Wald test p-value: grupos H60 80 vs H40 50
## DataFrame with 6 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE      stat    pvalue
##                 <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000000003  71.01158      -0.449500  1.611125 -0.278998  0.780247
## ENSG00000000419  57.54555      -0.107543  0.331128 -0.324780  0.745348
## ENSG00000000938   9.50582       1.518902  1.104914  1.374678  0.169231
## ENSG00000000971 241.68491       0.027187  0.263589  0.103142  0.917850
## ENSG00000001036  19.51369      -0.267467  1.193516 -0.224100  0.822680
## ENSG00000001084  41.83033      -0.420308  1.146995 -0.366443  0.714035
##                      padj
##                 <numeric>
## ENSG00000000003  0.996339
## ENSG00000000419  0.996339
## ENSG00000000938  0.869029
## ENSG00000000971  0.996339
## ENSG00000001036  0.996339
## ENSG00000001084  0.996339
```

```
# Los resultados se pueden ordenar para una mejor manejo de los datos.
Resultados_Ordenados <- Resultados_Age[order(Resultados_Age$padj),]
head(Resultados_Ordenados)
```

```
## log2 fold change (MLE): grupos H60 80 vs H40 50
## Wald test p-value: grupos H60 80 vs H40 50
## DataFrame with 6 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE      stat      pvalue
##                 <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000109511   81.9951       -9.82389   1.20809  -8.13177 4.23074e-16
## ENSG00000249948   81.6726       -9.81820   1.22435  -8.01912 1.06505e-15
## ENSG00000134463   61.9863       -9.42031   1.21503  -7.75318 8.96215e-15
## ENSG00000140505   84.5952       -9.86895   1.26804  -7.78283 7.09204e-15
## ENSG00000133027   55.2612       -9.25463   1.22635  -7.54649 4.47160e-14
## ENSG00000124588   53.7879       -9.21564   1.22516  -7.52200 5.39445e-14
##                       padj
##                  <numeric>
## ENSG00000109511 2.91879e-12
## ENSG00000249948 3.67390e-12
## ENSG00000134463 1.54575e-11
## ENSG00000140505 1.54575e-11
## ENSG00000133027 6.16991e-11
## ENSG00000124588 6.20272e-11
```

Tamaño muestral de genes diferencialmente expresados.

```
# Averiguamos cuantos genes hay diferencialmente expresados. En primer lugar,
# se ordenan los genes y omitimos los NAs.
Resultados_Ordenados <-Resultados_Ordenados[order(Resultados_Ordenados$padj,
                                                  na.last=NA),]

# Que escoja aquellos que sean inferior p-valor 0,05.
Genes_Dif_Age <- Resultados_Ordenados[Resultados_Ordenados$padj < 0.05, ]
dim(Genes_Dif_Age)
```

```
## [1] 58  6
```

```
# Podemos averiguar aquellos que tengan un padj inferior a 0.05 y ordenarlos
# según en up o down dependiendo de si logFC es negativo o positivo.
up <- rownames(Resultados_Ordenados)[Resultados_Ordenados$log2FoldChange >
                                       0 & Resultados_Ordenados$padj < 0.05]
down <- rownames(Resultados_Ordenados)[Resultados_Ordenados$log2FoldChange
                                     < 0 & Resultados_Ordenados$padj < 0.05]

# Número de genes con upregulación
length(up)
```

```
## [1] 6
```

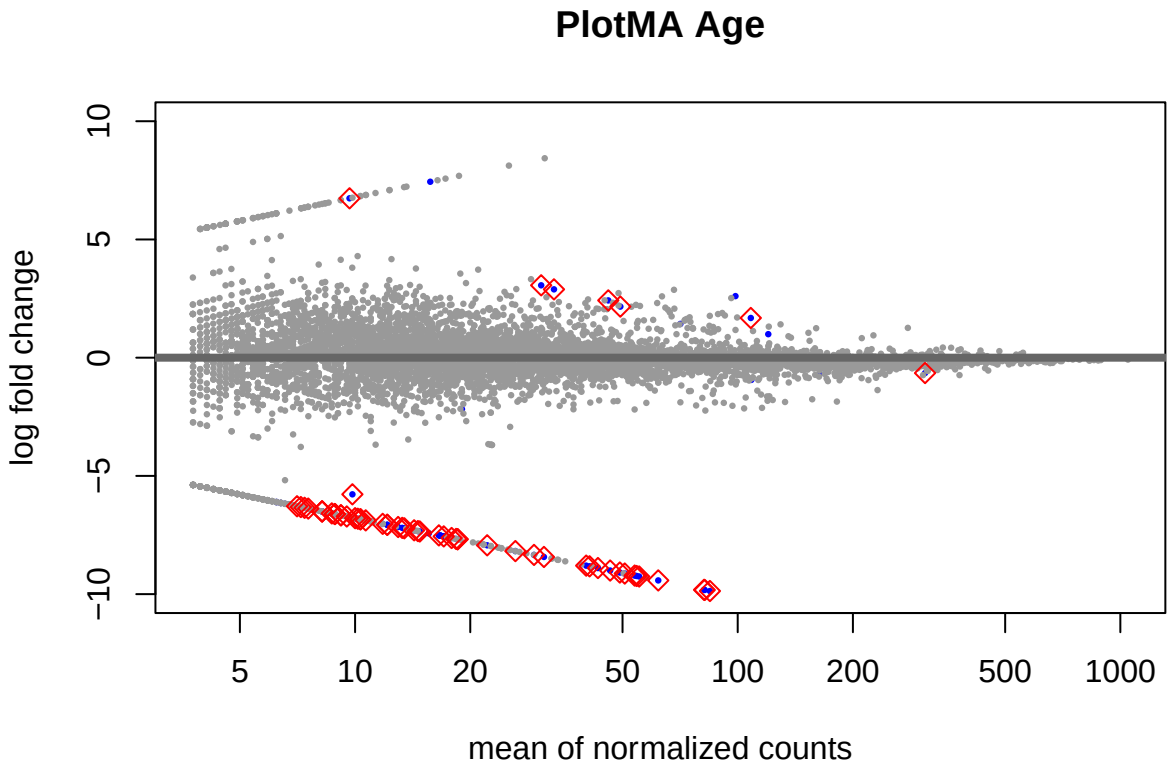
```r
# Número de genes downregulación
length(down)
```

```
## [1] 52
```

MA-plot, Volcanoplot y Heatmap

MA-plot

```r
# Una vez visto el número de genes, hace falta graficarlo.
plotMA(Resultados_Ordenados,ylim = c(-10, 10), main="PlotMA Age")

# Remarcar puntos con padj < 0.05
puntos_significativos <- Resultados_Ordenados$padj < 0.05

points(y=Resultados_Ordenados$log2FoldChange[puntos_significativos],
       x=Resultados_Ordenados$baseMean[puntos_significativos],
       col = "red", pch = 5)
```



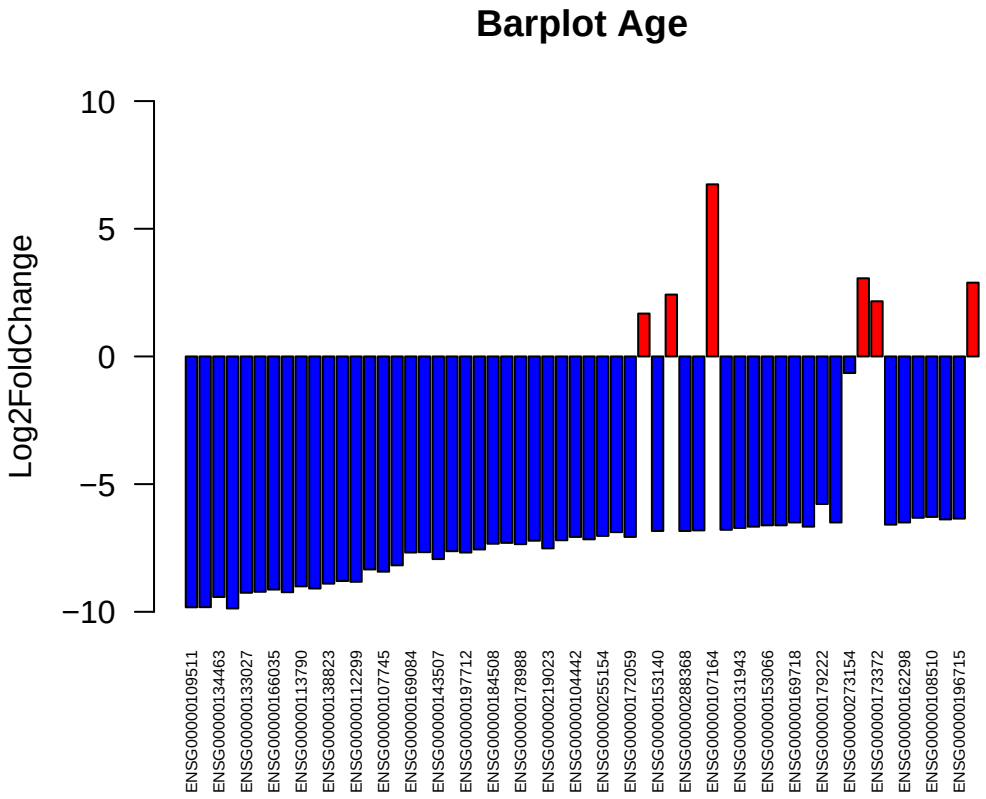
Volcano plot

```r
# Crear el gráfico de Volcano plot utilizando ggplot2.
Resultados_Ordenados_df<-data.frame(Resultados_Ordenados)
volcano <- ggplot(Resultados_Ordenados_df, aes(x = log2FoldChange, y =
                                                 -log10(pvalue))) +
# Creamos los distintos puntos negros con el color y la forma:
  geom_point(size = 1, color = "black") +
  geom_point(data = subset(Resultados_Ordenados_df, padj < 0.05), size = 1,
             color = "red") + # Puntos significativos en rojo
  labs(x = "Log Fold Change", y = "-log10(p-value)",
       title = "VolcanoPlot Age") + theme_minimal()

print(volcano)
```

Barplot

```r
colores_logFC <- ifelse(Genes_Dif_Age$log2FoldChange > 0, "red", "blue")
logFC <- Genes_Dif_Age$log2FoldChange

  # Colocamos los nombres de los genes en el eje Y
genes_age <- rownames(Genes_Dif_Age)
# Ajustamos la imagen
par(mar = c(5, 8, 4, 2) + 0.1)
# Crear un gráfico de barras
```

```
barplot(logFC, names.arg = genes_age, col = colores_logFC, ylim = c(-10,10) ,
        las = 2, ylab = "Log2FoldChange", cex.names = 0.5, main = "Barplot Age")
```



Panther.Database

```
# Exportamos los datos según las caracteristicas necesarias para que la página
# acepte nuestro documento.
Genes_Dif_Edad <- data.frame(GeneID = row.names(Genes_Dif_Age),
                             LogFC = Genes_Dif_Age$log2FoldChange)
write.table(x = Genes_Dif_Edad, file = "Genes_Dif_Age.txt",sep = "\t",
            quote = FALSE, row.names = FALSE)
```