# Datos Omicos PEC2

Eva Mª Ruiz Macias

6 de junio, 2020

## Table of Contents



## Introduccion

En este trabajo veremos como analizar los datos de conteo de RNA-seq usando el paquete R, y más concretamente edgeR. Los puntos que se tocaran van desde la lectura de los datos en R, hasta el control de calidad, la realización de análisis de expresión diferencial y pruebas de conjuntos de genes.

Los resultados de este trabajo se pueden encontrar en:

https://github.com/Tortufuriaperru/PEC2DatosOmicos.git

# Objetivos

Se analizaran los datos de expresion de tejido tiroideo de diferentes tipos: sin infiltración linfoidea, con pequeñas infiltraciones focales, y con infiltración linfoide extensa siguiendo los pasos mencionados anteriormente.

# Materiales y métodos

## Naturaleza de los datos

Para este trabajo contamos con dos archivos de datos llamados targets y counts que contienen la información de las muestras de un estudio obtenido del repositorio GTEx.

Dicho repositorio contiene datos de múltiples tipos en un total de 54 tejidos. En este trabajo utilizaremos los datos de expresión (RNA-seq) pertenecientes a un análisis del tiroides, en donde se compararan tres tipos de infiltración.

## Metodos para el analisis

### Identificación de grupos y quien pertenece a cada muestra.

En el archivo original contamos con 292 muestras de los siguientes tipos

- Not infiltrated tissues (NIT): 236 samples
- Small focal infiltrates (SFI): 42 samples
- Extensive lymphoid infiltrates (ELI): 14 samples.

Nos quedaremos con 10 muestras de cada grupo, que se mostraran posteriormente.

### Lectura de datos y selección de muestra

Procedemos a leer los archivos facilitados, y a seleccionar 10 muestras de cada tipo (30 en total):

```
targets <-  read.csv("C:/PEC2DatosOmicos/data/targets.csv", header = TRUE
)
counts <-  read.csv2("C:/PEC2DatosOmicos/data/counts.csv", header = TRUE,
sep = ";")
```

Seleccionamos la muestra de la siguiente forma:

```
set.seed(321, sample.kind = "Rounding")

## Warning in set.seed(321, sample.kind = "Rounding"): non-uniform 'Round
ing'
## sampler used

# muestras de tamaño 10 por grupo
```

```
targetsample <- targets%>%group_by(Group)%>%sample_n(size = 10, replace=F
)

# desactivo el paquete para que no me de problemas despues

detach("package:dplyr", unload = TRUE)

## Warning: 'dplyr' namespace cannot be unloaded:
##   namespace 'dplyr' is imported by 'BiocFileCache', 'dbplyr' so cannot
be unloaded

# nos quedamos con los elementos seleccionados

seleccion <- c(targetsample$Sample_Name)

# ahora nos quedamos con los elementos que ocupan la misma posicion en el
# archivo counts quitando la variable X
selectcounts <- counts[2:293][seleccion]
selectcounts <- subset(counts[2:293], select=seleccion)
# pasamos los nombres de los genes de la variable X a los nombres de las
filas
rownames(selectcounts) <- counts$X
# quitamos los puntos de los nombres de las columnas para su tratamiento
# posterior

rownames(selectcounts) <- gsub("\\..*", "", rownames(selectcounts),
                               fixed = FALSE)
head(rownames(selectcounts))

## [1] "ENSG00000223972" "ENSG00000227232" "ENSG00000243485" "ENSG0000023
7613"
## [5] "ENSG00000268020" "ENSG00000240361"

grupos <- rep(c("ELI", "NIT", "SFI"), each=10)

#head(selectcounts,3)
dim(selectcounts)

## [1] 56202    30
```

## Instalación de paquetes R

El analisis se ha hecho utilizando el programa R y los paquetes necesarios para dicho analisis son los siguientes:

require(knitr)

require(kableExtra)

require(ggplot2)

require(gplots)

require(limma)

require(Glimma)

require(edgeR)

require(stringr)

require(DESeq)

require(DESeq2)

require(RColorBrewer)

require(org.Hs.eg.db)

require(goseq)

require(GO.db)

require(dplyr)

## Formato de los datos

edgeR funciona con tablas de recuentos de lecturas de enteros, donde las filas correspondien a genes y las columnas a muestras independientes.

Se almacenaran los datos en un objeto de datos basado en listas llamado DGEList.

Este tipo de objeto es fácil de usar porque puede manipularse como cualquier lista en R.

```
grupos <- rep(c("ELI", "NIT", "SFI"), each=10)
# Creamos el objeto dGEList
dgList <- DGEList(selectcounts, group=grupos)
# Mostramos los datos
head(dgList, 2)
```

```
## An object of class "DGEList"
## $counts
##                 GTEX.ZYY3.1926.SM.5GZXS GTEX.YJ89.0726.SM.5P9F7
## ENSG00000223972                       6                       4
## ENSG00000227232                    1003                    1325
##                 GTEX.11XUK.0226.SM.5EQLW GTEX.YFC4.2626.SM.5P9FQ
## ENSG00000223972                        0                       1
## ENSG00000227232                      419                    1472
##                 GTEX.13NZ9.1126.SM.5MR37 GTEX.R55G.0726.SM.2TC6J
## ENSG00000223972                        0                       3
## ENSG00000227232                     1002                     134
##                 GTEX.PLZ4.1226.SM.2I5FE GTEX.TMMY.0826.SM.33HB9
## ENSG00000223972                       5                       3
```

```
## ENSG00000227232                      489                       979
##                  GTEX.14AS3.0226.SM.5Q5B6 GTEX.13QJC.0826.SM.5RQKC
## ENSG00000223972                        0                         0
## ENSG00000227232                      834                       825
##                  GTEX.QV31.0726.SM.3GAEG GTEX.13OW7.0826.SM.5L3EL
## ENSG00000223972                        3                         0
## ENSG00000227232                      450                       629
##                  GTEX.X8HC.0726.SM.46MWG GTEX.11DXX.0226.SM.5P9HL
## ENSG00000223972                        0                         4
## ENSG00000227232                      879                       825
##                  GTEX.Q734.0526.SM.2I3EH GTEX.13113.0126.SM.5LZVX
## ENSG00000223972                        2                         1
## ENSG00000227232                      749                       687
##                  GTEX.R3RS.0726.SM.3GIJR GTEX.13S86.1126.SM.5RQJX
## ENSG00000223972                        1                         1
## ENSG00000227232                      176                       800
##                  GTEX.13FTY.0726.SM.5J2OH GTEX.ZYFC.0926.SM.5GZWW
## ENSG00000223972                        1                         1
## ENSG00000227232                      675                      1051
##                  GTEX.QLQ7.0726.SM.2I5G2 GTEX.ZLV1.0126.SM.4WWBZ
## ENSG00000223972                        6                         2
## ENSG00000227232                      666                       689
##                  GTEX.Y5V6.0526.SM.4VBRV GTEX.13FH7.0126.SM.5KLZ1
## ENSG00000223972                        3                         5
## ENSG00000227232                      482                       576
##                  GTEX.13NZ8.0226.SM.5J2OK GTEX.R55C.0626.SM.2TF4Q
## ENSG00000223972                        1                         9
## ENSG00000227232                     1164                       302
##                  GTEX.WYVS.0326.SM.3NM9V GTEX.131YS.0726.SM.5P9G9
## ENSG00000223972                        6                         1
## ENSG00000227232                      820                      1487
##                  GTEX.11GS4.0826.SM.5986J GTEX.13FXS.0726.SM.5LZXJ
## ENSG00000223972                        0                         5
## ENSG00000227232                      533                      1564
##
## $samples
##                            group lib.size norm.factors
## GTEX.ZYY3.1926.SM.5GZXS      ELI 48915857            1
## GTEX.YJ89.0726.SM.5P9F7      ELI 73988083            1
## GTEX.11XUK.0226.SM.5EQLW     ELI 50019489            1
## GTEX.YFC4.2626.SM.5P9FQ      ELI 81226878            1
## GTEX.13NZ9.1126.SM.5MR37     ELI 61447691            1
## 25 more rows ...

names(dgList)

## [1] "counts"  "samples"

dgList$samples
```

```
##                           group lib.size norm.factors
## GTEX.ZYY3.1926.SM.5GZXS     ELI 48915857            1
## GTEX.YJ89.0726.SM.5P9F7     ELI 73988083            1
## GTEX.11XUK.0226.SM.5EQLW    ELI 50019489            1
## GTEX.YFC4.2626.SM.5P9FQ     ELI 81226878            1
## GTEX.13NZ9.1126.SM.5MR37    ELI 61447691            1
## GTEX.R55G.0726.SM.2TC6J     ELI 15483883            1
## GTEX.PLZ4.1226.SM.2I5FE     ELI 64441734            1
## GTEX.TMMY.0826.SM.33HB9     ELI 85633787            1
## GTEX.14AS3.0226.SM.5Q5B6    ELI 42011392            1
## GTEX.13QJC.0826.SM.5RQKC    ELI 48836801            1
## GTEX.QV31.0726.SM.3GAEG     NIT 49650895            1
## GTEX.13OW7.0826.SM.5L3EL    NIT 41666882            1
## GTEX.X8HC.0726.SM.46MWG     NIT 50137652            1
## GTEX.11DXX.0226.SM.5P9HL    NIT 85676907            1
## GTEX.Q734.0526.SM.2I3EH     NIT 50362382            1
## GTEX.13113.0126.SM.5LZVX    NIT 43630813            1
## GTEX.R3RS.0726.SM.3GIJR     NIT 12431887            1
## GTEX.13S86.1126.SM.5RQJX    NIT 40167105            1
## GTEX.13FTY.0726.SM.5J2OH    NIT 58965885            1
## GTEX.ZYFC.0926.SM.5GZWW     NIT 51417663            1
## GTEX.QLQ7.0726.SM.2I5G2     SFI 84712651            1
## GTEX.ZLV1.0126.SM.4WWBZ     SFI 55426907            1
## GTEX.Y5V6.0526.SM.4VBRV     SFI 68714782            1
## GTEX.13FH7.0126.SM.5KLZ1    SFI 66583792            1
## GTEX.13NZ8.0226.SM.5J2OK    SFI 59535746            1
## GTEX.R55C.0626.SM.2TF4Q     SFI 39862745            1
## GTEX.WYVS.0326.SM.3NM9V     SFI 76726397            1
## GTEX.131YS.0726.SM.5P9G9    SFI 67931798            1
## GTEX.11GS4.0826.SM.5986J    SFI 50383412            1
## GTEX.13FXS.0726.SM.5LZXJ    SFI 47570049            1
```

## Filtrado para eliminar genes poco expresados

Los genes con recuentos muy bajos en todas las bibliotecas proporcionan poca evidencia de expresión diferencial e interfieren con algunas de las aproximaciones estadísticas que se utilizaran más adelante. También se suman a la carga de las pruebas múltiples al estimar las tasas de falsas, reduciendo el poder de detectar genes expresados diferencialmente. Estos genes deben filtrarse antes de un análisis posterior.

Hay algunas formas de filtrar los genes poco expresados. En este conjunto de datos, elegimos retener genes si se expresan en un recuento por millón (CPM) superior a 0,5 en al menos dos muestras.

Utilizaremos la función cpm de la biblioteca edgeR para generar los valores de CPM y luego filtrarlos. Hay que tener en cuenta que al convertir a CPM estamos normalizando las diferentes profundidades de secuencia para cada muestra.

```
countsPerMillion <- cpm(dgList)
summary(countsPerMillion)

##  GTEX.ZYY3.1926.SM.5GZXS GTEX.YJ89.0726.SM.5P9F7 GTEX.11XUK.0226.SM.5E
QLW
##  Min.   :    0.000       Min.   :    0.000       Min.   :    0.000
##  1st Qu.:    0.000       1st Qu.:    0.000       1st Qu.:    0.000
##  Median :    0.020       Median :    0.027       Median :    0.040
##  Mean   :   17.793       Mean   :   17.793       Mean   :   17.793
##  3rd Qu.:    3.128       3rd Qu.:    3.730       3rd Qu.:    3.459
##  Max.   :27235.381       Max.   :24165.486       Max.   :21323.908
##  GTEX.YFC4.2626.SM.5P9FQ GTEX.13NZ9.1126.SM.5MR37 GTEX.R55G.0726.SM.2T
C6J
##  Min.   :    0.000       Min.   :    0.000        Min.   :    0.000
##  1st Qu.:    0.000       1st Qu.:    0.000        1st Qu.:    0.000
##  Median :    0.037       Median :    0.033        Median :    0.000
##  Mean   :   17.793       Mean   :   17.793        Mean   :   17.793
##  3rd Qu.:    4.050       3rd Qu.:    3.873        3rd Qu.:    3.746
##  Max.   :12196.443       Max.   :10679.734        Max.   :24049.135
##  GTEX.PLZ4.1226.SM.2I5FE GTEX.TMMY.0826.SM.33HB9 GTEX.14AS3.0226.SM.5Q
5B6
##  Min.   :    0.000       Min.   :    0.00        Min.   :    0.00
##  1st Qu.:    0.000       1st Qu.:    0.00        1st Qu.:    0.00
##  Median :    0.031       Median :    0.04        Median :    0.05
##  Mean   :   17.793       Mean   :   17.79        Mean   :   17.79
##  3rd Qu.:    2.917       3rd Qu.:    2.80        3rd Qu.:    3.71
##  Max.   :29715.308       Max.   :48667.11        Max.   :33592.56
##  GTEX.13QJC.0826.SM.5RQKC GTEX.QV31.0726.SM.3GAEG GTEX.13OW7.0826.SM.5
L3EL
##  Min.   :    0.000        Min.   :    0.00        Min.   :    0.000
##  1st Qu.:    0.000        1st Qu.:    0.00        1st Qu.:    0.000
##  Median :    0.041        Median :    0.02        Median :    0.024
##  Mean   :   17.793        Mean   :   17.79        Mean   :   17.793
##  3rd Qu.:    3.215        3rd Qu.:    2.26        3rd Qu.:    2.928
##  Max.   :28440.253        Max.   :41245.10        Max.   :21764.816
##  GTEX.X8HC.0726.SM.46MWG GTEX.11DXX.0226.SM.5P9HL GTEX.Q734.0526.SM.2I
3EH
##  Min.   :    0.000       Min.   :    0.00         Min.   :    0.000
##  1st Qu.:    0.000       1st Qu.:    0.00         1st Qu.:    0.000
##  Median :    0.020       Median :    0.02         Median :    0.040
##  Mean   :   17.793       Mean   :   17.79         Mean   :   17.793
##  3rd Qu.:    2.847       3rd Qu.:    2.11         3rd Qu.:    3.058
##  Max.   :13297.811       Max.   :36450.10         Max.   :24868.562
##  GTEX.13113.0126.SM.5LZVX GTEX.R3RS.0726.SM.3GIJR GTEX.13S86.1126.SM.5
RQJX
##  Min.   :    0.000        Min.   :    0.000       Min.   :    0.000
##  1st Qu.:    0.000        1st Qu.:    0.000       1st Qu.:    0.000
##  Median :    0.023        Median :    0.000       Median :    0.000
##  Mean   :   17.793        Mean   :   17.793       Mean   :   17.793
##  3rd Qu.:    2.842        3rd Qu.:    2.896       3rd Qu.:    2.415
```

```
##  Max.   :27584.817        Max.   :28143.515       Max.   :29716.057
##  GTEX.13FTY.0726.SM.5J2OH GTEX.ZYFC.0926.SM.5GZWW GTEX.QLQ7.0726.SM.2I
5G2
##  Min.   :    0.000        Min.   :    0.00        Min.   :    0.00
##  1st Qu.:    0.000        1st Qu.:    0.00        1st Qu.:    0.00
##  Median :    0.034        Median :    0.02        Median :    0.02
##  Mean   :   17.793        Mean   :   17.79        Mean   :   17.79
##  3rd Qu.:    3.053        3rd Qu.:    2.65        3rd Qu.:    2.35
##  Max.   :26554.744        Max.   :44670.33        Max.   :49250.75
##  GTEX.ZLV1.0126.SM.4WWBZ GTEX.Y5V6.0526.SM.4VBRV GTEX.13FH7.0126.SM.5K
LZ1
##  Min.   :    0.000       Min.   :    0.00        Min.   :    0.000
##  1st Qu.:    0.000       1st Qu.:    0.00        1st Qu.:    0.000
##  Median :    0.036       Median :    0.01        Median :    0.030
##  Mean   :   17.793       Mean   :   17.79        Mean   :   17.793
##  3rd Qu.:    3.248       3rd Qu.:    2.26        3rd Qu.:    2.628
##  Max.   :19982.389       Max.   :49670.00        Max.   :31423.894
##  GTEX.13NZ8.0226.SM.5J2OK GTEX.R55C.0626.SM.2TF4Q GTEX.WYVS.0326.SM.3N
M9V
##  Min.   :    0.000        Min.   :    0.000       Min.   :    0.000
##  1st Qu.:    0.000        1st Qu.:    0.000       1st Qu.:    0.000
##  Median :    0.017        Median :    0.025       Median :    0.026
##  Mean   :   17.793        Mean   :   17.793       Mean   :   17.793
##  3rd Qu.:    2.839        3rd Qu.:    2.659       3rd Qu.:    3.089
##  Max.   :13038.251        Max.   :31177.532       Max.   :28452.659
##  GTEX.131YS.0726.SM.5P9G9 GTEX.11GS4.0826.SM.5986J GTEX.13FXS.0726.SM.
5LZXJ
##  Min.   :    0.00         Min.   :    0.00         Min.   :    0.000
##  1st Qu.:    0.00         1st Qu.:    0.00         1st Qu.:    0.000
##  Median :    0.03         Median :    0.04         Median :    0.021
##  Mean   :   17.79         Mean   :   17.79         Mean   :   17.793
##  3rd Qu.:    3.47         3rd Qu.:    2.64         3rd Qu.:    3.195
##  Max.   :34335.82         Max.   :31870.83         Max.   :18527.267
```

```
# valores mayores que 0.5
countCheck <- countsPerMillion > 0.5
# Esto produce una salida con valores logicos TRUEs y FALSEs
head(countCheck, 2)
```

```
##                 GTEX.ZYY3.1926.SM.5GZXS GTEX.YJ89.0726.SM.5P9F7
## ENSG00000223972                   FALSE                   FALSE
## ENSG00000227232                    TRUE                    TRUE
##                 GTEX.11XUK.0226.SM.5EQLW GTEX.YFC4.2626.SM.5P9FQ
## ENSG00000223972                    FALSE                   FALSE
## ENSG00000227232                     TRUE                    TRUE
##                 GTEX.13NZ9.1126.SM.5MR37 GTEX.R55G.0726.SM.2TC6J
## ENSG00000223972                    FALSE                   FALSE
## ENSG00000227232                     TRUE                    TRUE
##                 GTEX.PLZ4.1226.SM.2I5FE GTEX.TMMY.0826.SM.33HB9
## ENSG00000223972                   FALSE                   FALSE
```

```
## ENSG00000227232                    TRUE                     TRUE
##                 GTEX.14AS3.0226.SM.5Q5B6 GTEX.13QJC.0826.SM.5RQKC
## ENSG00000223972                   FALSE                    FALSE
## ENSG00000227232                    TRUE                     TRUE
##                 GTEX.QV31.0726.SM.3GAEG GTEX.13OW7.0826.SM.5L3EL
## ENSG00000223972                   FALSE                    FALSE
## ENSG00000227232                    TRUE                     TRUE
##                 GTEX.X8HC.0726.SM.46MWG GTEX.11DXX.0226.SM.5P9HL
## ENSG00000223972                   FALSE                    FALSE
## ENSG00000227232                    TRUE                     TRUE
##                 GTEX.Q734.0526.SM.2I3EH GTEX.13113.0126.SM.5LZVX
## ENSG00000223972                   FALSE                    FALSE
## ENSG00000227232                    TRUE                     TRUE
##                 GTEX.R3RS.0726.SM.3GIJR GTEX.13S86.1126.SM.5RQJX
## ENSG00000223972                   FALSE                    FALSE
## ENSG00000227232                    TRUE                     TRUE
##                 GTEX.13FTY.0726.SM.5J2OH GTEX.ZYFC.0926.SM.5GZWW
## ENSG00000223972                    FALSE                    FALSE
## ENSG00000227232                     TRUE                     TRUE
##                 GTEX.QLQ7.0726.SM.2I5G2 GTEX.ZLV1.0126.SM.4WWBZ
## ENSG00000223972                   FALSE                   FALSE
## ENSG00000227232                    TRUE                    TRUE
##                 GTEX.Y5V6.0526.SM.4VBRV GTEX.13FH7.0126.SM.5KLZ1
## ENSG00000223972                   FALSE                    FALSE
## ENSG00000227232                    TRUE                     TRUE
##                 GTEX.13NZ8.0226.SM.5J2OK GTEX.R55C.0626.SM.2TF4Q
## ENSG00000223972                    FALSE                    FALSE
## ENSG00000227232                     TRUE                     TRUE
##                 GTEX.WYVS.0326.SM.3NM9V GTEX.131YS.0726.SM.5P9G9
## ENSG00000223972                   FALSE                    FALSE
## ENSG00000227232                    TRUE                     TRUE
##                 GTEX.11GS4.0826.SM.5986J GTEX.13FXS.0726.SM.5LZXJ
## ENSG00000223972                    FALSE                    FALSE
## ENSG00000227232                     TRUE                     TRUE

# Cuantos trues hay en cada fila
table(rowSums(countCheck))

##
##     0     1     2     3     4     5     6     7     8     9    10    1
1    12
## 32941  1259   658   474   361   322   259   235   250   208   215   18
8   182
##    13    14    15    16    17    18    19    20    21    22    23    2
4    25
##   184   184   201   180   149   154   181   157   171   158   193   21
2   172
##    26    27    28    29    30
##   223   268   355   573 14935
```

```r
# Nos quedamos con los que tengan al menos 2 TRUES
keep <- which(rowSums(countCheck) >= 2)
dgList <- dgList[keep,]
summary(cpm(dgList))
```

```
##  GTEX.ZYY3.1926.SM.5GZXS GTEX.YJ89.0726.SM.5P9F7 GTEX.11XUK.0226.SM.5E
QLW
##  Min.   :    0.000       Min.   :    0.000       Min.   :    0.00
##  1st Qu.:    1.083       1st Qu.:    1.284       1st Qu.:    1.22
##  Median :   10.610       Median :   11.867       Median :   11.02
##  Mean   :   45.418       Mean   :   45.414       Mean   :   45.41
##  3rd Qu.:   40.105       3rd Qu.:   43.575       3rd Qu.:   40.82
##  Max.   :27235.381       Max.   :24165.486       Max.   :21323.91
##  GTEX.YFC4.2626.SM.5P9FQ GTEX.13NZ9.1126.SM.5MR37 GTEX.R55G.0726.SM.2T
C6J
##  Min.   :    0.000       Min.   :    0.000        Min.   :    0.000
##  1st Qu.:    1.465       1st Qu.:    1.334        1st Qu.:    1.356
##  Median :   12.453       Median :   12.352        Median :   11.560
##  Mean   :   45.404       Mean   :   45.413        Mean   :   45.405
##  3rd Qu.:   44.366       3rd Qu.:   46.295        3rd Qu.:   42.237
##  Max.   :12196.443       Max.   :10679.734        Max.   :24049.135
##  GTEX.PLZ4.1226.SM.2I5FE GTEX.TMMY.0826.SM.33HB9 GTEX.14AS3.0226.SM.5Q
5B6
##  Min.   :    0.000       Min.   :    0.00        Min.   :    0.00
##  1st Qu.:    1.013       1st Qu.:    0.97        1st Qu.:    1.29
##  Median :    9.846       Median :    9.63        Median :   11.95
##  Mean   :   45.415       Mean   :   45.41        Mean   :   45.41
##  3rd Qu.:   40.657       3rd Qu.:   38.56        3rd Qu.:   43.13
##  Max.   :29715.308       Max.   :48667.11        Max.   :33592.56
##  GTEX.13QJC.0826.SM.5RQKC GTEX.QV31.0726.SM.3GAEG GTEX.13OW7.0826.SM.5
L3EL
##  Min.   :    0.000        Min.   :    0.00        Min.   :    0.000
##  1st Qu.:    1.126        1st Qu.:    0.79        1st Qu.:    1.038
##  Median :   10.402        Median :    9.10        Median :   10.104
##  Mean   :   45.415        Mean   :   45.42        Mean   :   45.413
##  3rd Qu.:   40.584        3rd Qu.:   39.35        3rd Qu.:   41.376
##  Max.   :28440.253        Max.   :41245.10        Max.   :21764.816
##  GTEX.X8HC.0726.SM.46MWG GTEX.11DXX.0226.SM.5P9HL GTEX.Q734.0526.SM.2I
3EH
##  Min.   :    0.000       Min.   :    0.00         Min.   :    0.000
##  1st Qu.:    0.977       1st Qu.:    0.70         1st Qu.:    1.033
##  Median :    9.733       Median :    8.53         Median :   10.831
##  Mean   :   45.416       Mean   :   45.42         Mean   :   45.415
##  3rd Qu.:   41.785       3rd Qu.:   38.41         3rd Qu.:   41.435
##  Max.   :13297.811       Max.   :36450.10         Max.   :24868.562
##  GTEX.13113.0126.SM.5LZVX GTEX.R3RS.0726.SM.3GIJR GTEX.13S86.1126.SM.5
RQJX
##  Min.   :    0.000        Min.   :    0.000       Min.   :    0.000
##  1st Qu.:    1.008        1st Qu.:    1.046       1st Qu.:    0.797
##  Median :    9.901        Median :   10.055       Median :    9.460
```

```
##  Mean   :   45.416        Mean   :   45.410       Mean   :   45.426
##  3rd Qu.:   41.072        3rd Qu.:   40.541       3rd Qu.:   39.510
##  Max.   :27584.817        Max.   :28143.515       Max.   :29716.057
##  GTEX.13FTY.0726.SM.5J2OH GTEX.ZYFC.0926.SM.5GZWW GTEX.QLQ7.0726.SM.2I
5G2
##  Min.   :    0.000        Min.   :    0.00        Min.   :    0.00
##  1st Qu.:    1.034        1st Qu.:    0.91        1st Qu.:    0.81
##  Median :   10.710        Median :    9.56        Median :    8.91
##  Mean   :   45.417        Mean   :   45.42        Mean   :   45.42
##  3rd Qu.:   41.834        3rd Qu.:   41.39        3rd Qu.:   39.07
##  Max.   :26554.744        Max.   :44670.33        Max.   :49250.75
##  GTEX.ZLV1.0126.SM.4WWBZ GTEX.Y5V6.0526.SM.4VBRV GTEX.13FH7.0126.SM.5K
LZ1
##  Min.   :    0.000       Min.   :    0.00        Min.   :    0.000
##  1st Qu.:    1.083       1st Qu.:    0.76        1st Qu.:    0.916
##  Median :   11.240       Median :    8.95        Median :    9.529
##  Mean   :   45.416       Mean   :   45.43        Mean   :   45.422
##  3rd Qu.:   42.069       3rd Qu.:   39.42        3rd Qu.:   39.589
##  Max.   :19982.389       Max.   :49670.00        Max.   :31423.894
##  GTEX.13NZ8.0226.SM.5J2OK GTEX.R55C.0626.SM.2TF4Q GTEX.WYVS.0326.SM.3N
M9V
##  Min.   :    0.000        Min.   :    0.000       Min.   :    0.000
##  1st Qu.:    0.957        1st Qu.:    0.928       1st Qu.:    1.069
##  Median :   10.221        Median :    9.708       Median :   10.518
##  Mean   :   45.418        Mean   :   45.416       Mean   :   45.414
##  3rd Qu.:   43.419        3rd Qu.:   41.267       3rd Qu.:   42.358
##  Max.   :13038.251        Max.   :31177.532       Max.   :28452.659
##  GTEX.131YS.0726.SM.5P9G9 GTEX.11GS4.0826.SM.5986J GTEX.13FXS.0726.SM.
5LZXJ
##  Min.   :    0.00         Min.   :    0.00         Min.   :    0.000
##  1st Qu.:    1.21         1st Qu.:    0.99         1st Qu.:    1.051
##  Median :   10.94         Median :    9.23         Median :   11.173
##  Mean   :   45.42         Mean   :   45.42         Mean   :   45.422
##  3rd Qu.:   40.91         3rd Qu.:   39.79         3rd Qu.:   42.669
##  Max.   :34335.82         Max.   :31870.83         Max.   :18527.267
```

```
dim(dgList)
```

```
## [1] 22002    30
```

Esto reduce el conjunto de datos de 56202 genes a 22002. Para los genes filtrados, hay muy poca potencia para detectar la expresión diferencial, por lo que el filtrado pierde poca información.

## Control de calidad

Una vez filtrados los genes con poca expresión y almacenados los datos en el objeto que hemos creado, veamos la calidad de los datos.

Primero, podemos verificar cuántas lecturas tenemos para cada muestra:

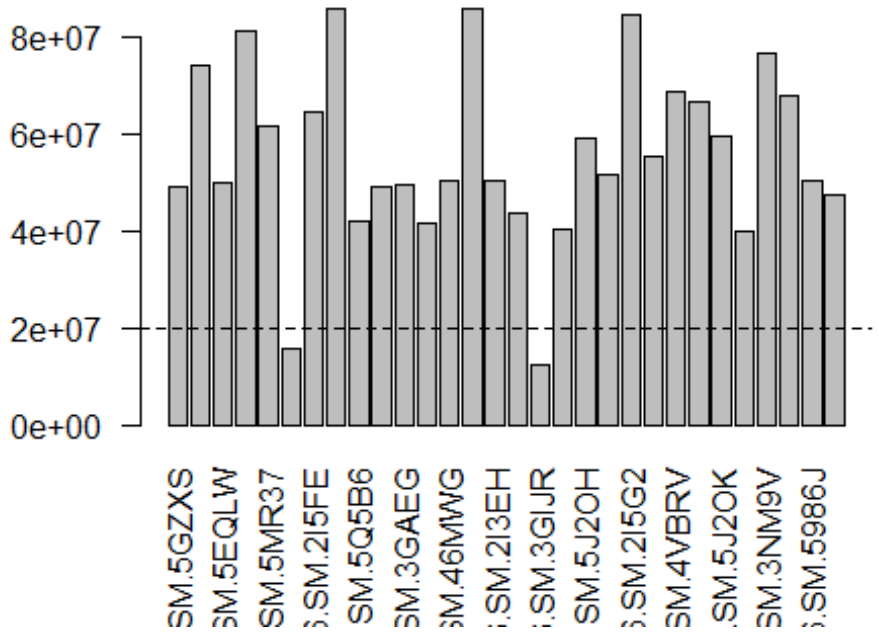
```
dgList$samples$lib.size
```

```
##  [1] 48915857 73988083 50019489 81226878 61447691 15483883 64441734 85
633787
##  [9] 42011392 48836801 49650895 41666882 50137652 85676907 50362382 43
630813
## [17] 12431887 40167105 58965885 51417663 84712651 55426907 68714782 66
583792
## [25] 59535746 39862745 76726397 67931798 50383412 47570049
```

Hay que tener en cuenta que el "size factor" de DSeq no es igual que "norm factor" de edgeR.

También podemos trazar los tamaños de la biblioteca como un diagrama de barras para ver si hay más discrepancias entre las muestras más fácilmente

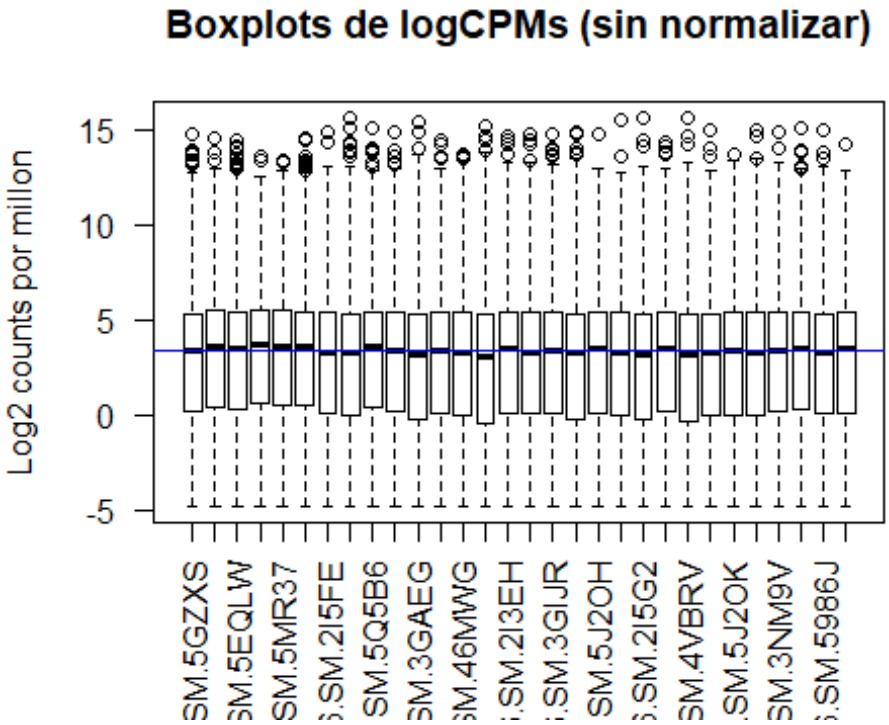
```
barplot(dgList$samples$lib.size, names=colnames(dgList), las=2)
abline(h=20e6, lty=2)
```




Los datos de recuento no se distribuyen normalmente. Vamos a hacer diagramas de cajas para verificar la distribución de los recuentos de lectura en la escala log2. Podemos usar la funcion cpm para obtener recuentos de log2 por millón, que se corrigen para los diferentes tamaños de biblioteca. La funcion cpmf también agrega un pequeño desplazamiento para evitar tomar el registro de cero.

```
boxplot(cpm(dgList, log = TRUE), xlab="", ylab="Log2 counts por millon",l
as=2)
```

```r
# Añadimos la mediana logCPM en color azul
abline(h=median(cpm(dgList, log = TRUE)),col="blue")
title("Boxplots de logCPMs (sin normalizar)")
```

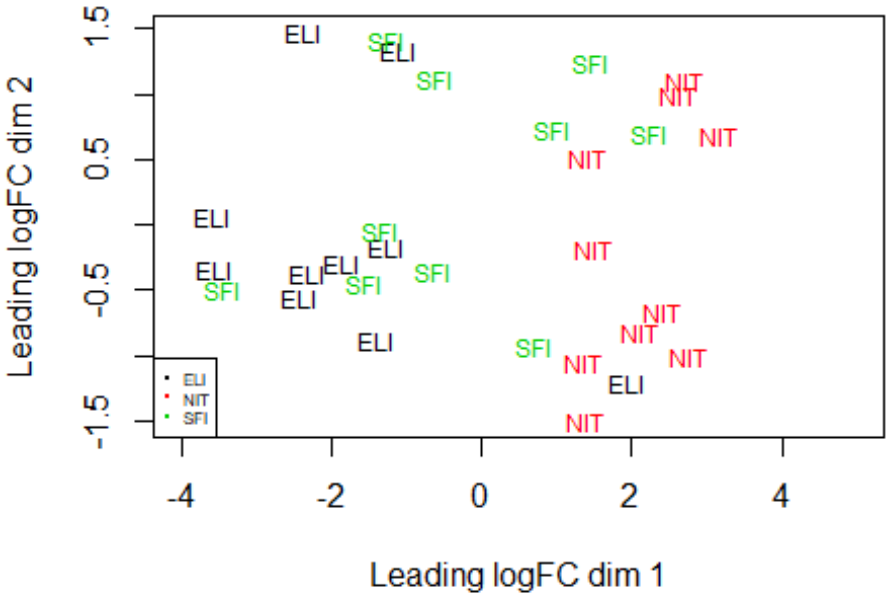



De los diagramas de caja vemos que, en general, las distribuciones de densidad de las intensidades logarítmicas en bruto no son idénticas pero tampoco muy diferentes.

Un MDSplot es una visualización de un análisis de componentes principales, que determina las mayores fuentes de variación en los datos.Muestra distancias, en términos de coeficiente de variación biológica (BCV), entre muestras.

Un análisis de componentes principales es un ejemplo de un análisis no supervisado, donde no necesitamos especificar los grupos. Si el experimento está bien controlado y ha funcionado bien, lo que esperamos ver es que las mayores fuentes de variación en los datos son los grupos en los que estamos interesados. También es una herramienta muy útil para el control de calidad y la comprobación de valores atípicos. Podemos usar la funcion plotMDS para crear el diagrama MDS.

```r
plotMDS(dgList, labels=dgList$samples$group,
        cex=0.75,
        xlim=c(-4, 5),
        col=as.numeric(dgList$samples$group))
legend("bottomleft", as.character(unique(dgList$samples$group)),
       col=1:3,
       pch=20,
       cex = 0.5)
```




Otra alternativa es generar un diagrama MDS interactivo utilizando el paquete Glimma . Esto permite explorar interactivamente las diferentes dimensiones.

```
glMDSPlot(dgList, groups=grupos, folder="mds")
```

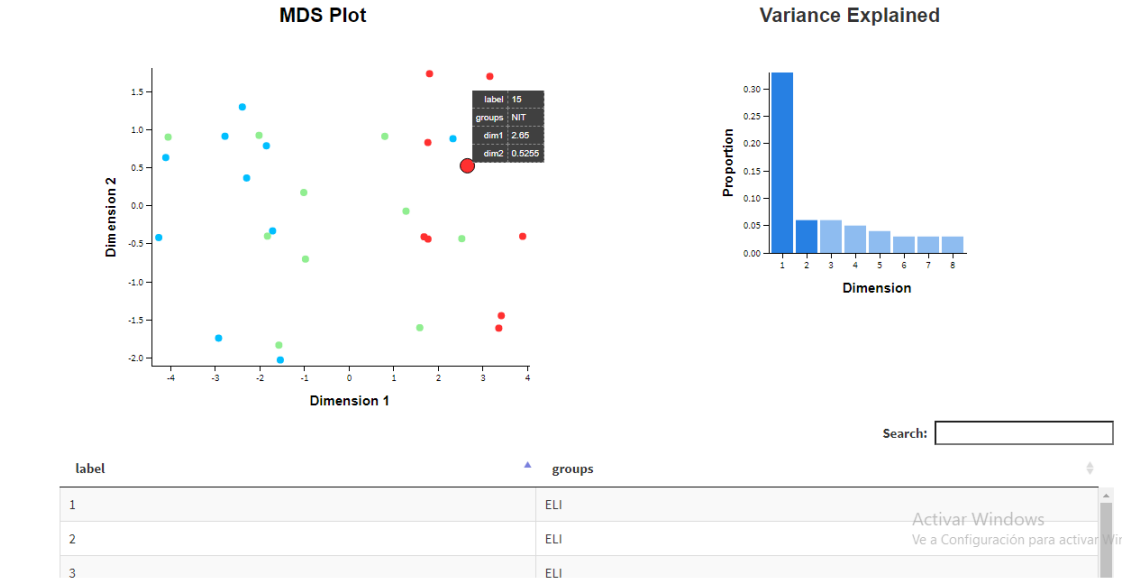



*Diagrama MDS interactivo*

La salida de glMDSPlot es una página html que muestra el diagrama MDS a la izquierda y la cantidad de variación explicada por cada dimensión en un diagrama de

barras a la derecha. Podemos desplazarnos sobre los puntos para encontrar información de la muestra y cambiar entre dimensiones sucesivas en el diagrama MDS haciendo clic en las barras del diagrama de barras.

```
logcounts <- cpm(dgList, log=TRUE)
```

## Normalización de los datos

La función calcNormFactors de edgeR calcula los factores de normalización entre bibliotecas.

```
dgList <- calcNormFactors(dgList, method="TMM")
```

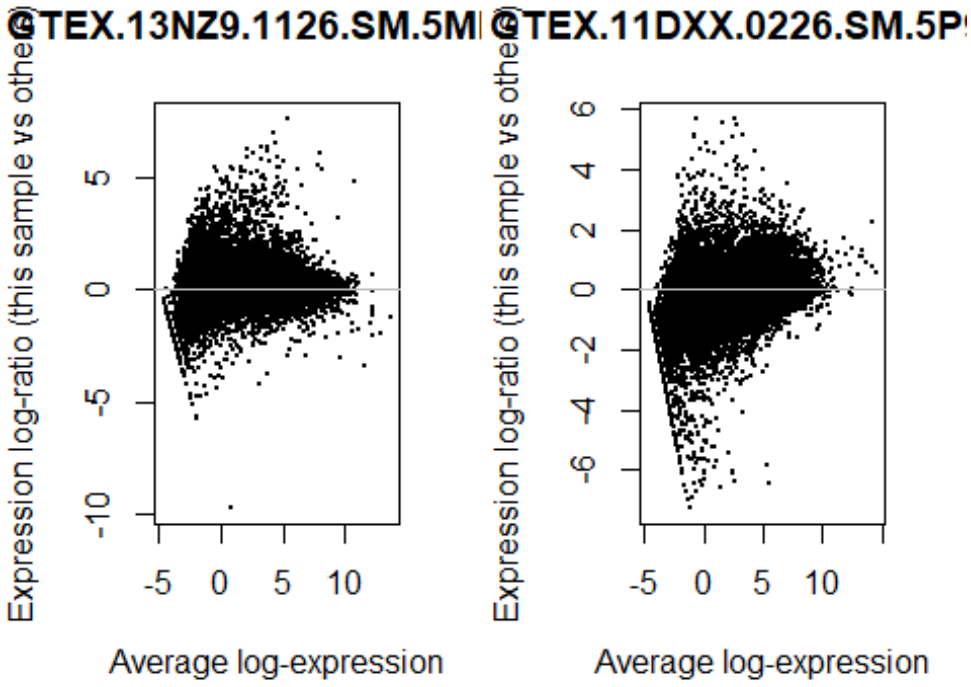
Esto actualizará los factores de normalización del objeto dgList (sus valores predeterminados son 1). veamos los factores de normalización para las muestras.

Los factores de normalización multiplican a la unidad en todas las bibliotecas. Un factor de normalización por debajo de uno indica que el tamaño de la biblioteca se reducirá, ya que hay más supresión (es decir, sesgo de composición) en esa biblioteca en relación con las otras bibliotecas. Esto también es equivalente a escalar los recuentos hacia arriba en esa muestra. Por el contrario, un factor superior a uno aumenta el tamaño de la biblioteca y es equivalente a reducir los recuentos.

La muestra GTEX.11DXX.0226.SM.5P9HL tiene el factor de normalización más pequeño, y GTEX.13NZ9.1126.SM.5MR37 tiene el más grande. Si trazamos gráficas de diferencia de medias usando la función plotMD para estas muestras, deberíamos poder ver el problema de sesgo de composición. Usaremos el logcounts, que se ha normalizado para el tamaño de la biblioteca, pero no para el sesgo de composición.

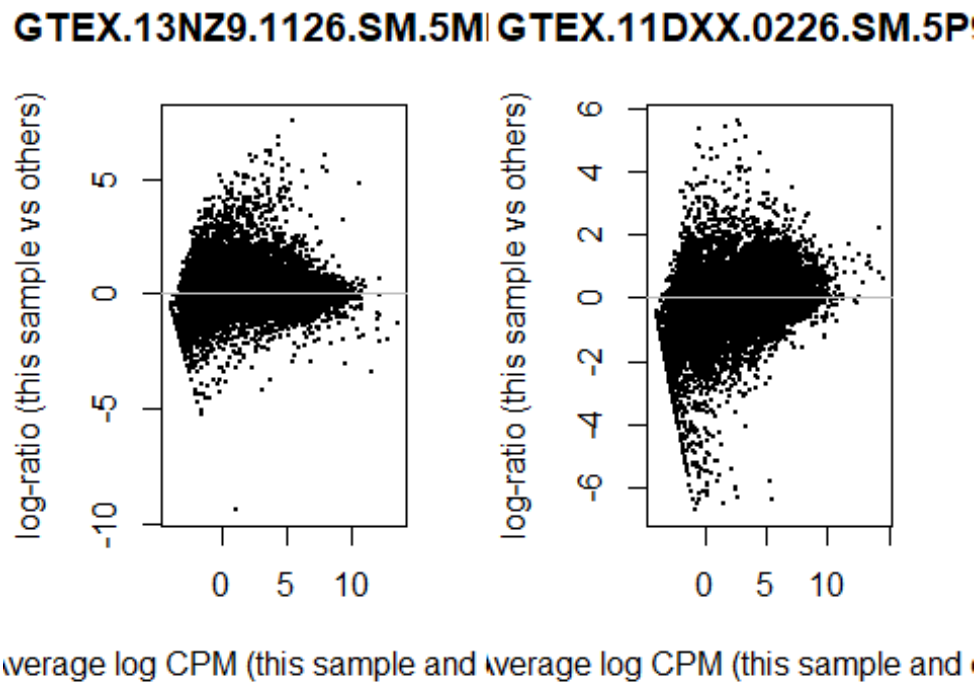
```
logcounts <- cpm(dgList,log=TRUE)

par(mfrow=c(1,2))
plotMD(logcounts,column = 5)
abline(h=0,col="grey")
plotMD(logcounts,column = 14)
abline(h=0,col="grey")
```




Los gráficos de diferencia de medias muestran la expresión promedio (media: eje x) frente a log-fold-changes (diferencia: eje y). Veamos las graficas con dgList:

```
par(mfrow=c(1,2))
plotMD(dgList,column = 5)
abline(h=0,col="grey")
plotMD(dgList,column = 14)
abline(h=0,col="grey")
```




```
save(grupos,dgList,file="C:/Pec2DatosOmicos/results/preprocessing.Rdata")
```

## Expresión diferencial

### Estimación de la dispersión

Un paso importante en el análisis de los datos DGE utilizando el modelo NB es estimar el parámetro de dispersión para cada etiqueta, una medida del grado de variación entre bibliotecas. La estimación de la dispersión común da una idea de la variabilidad general a través del genoma para el conjunto de datos.

Aquí vamos a hacer la estimación suponiendo que todo tiene la misma dispersión común:

```
d1 <- estimateCommonDisp(dgList, verbose=T)

## Disp = 0.2448 , BCV = 0.4948

names(d1)

## [1] "counts"            "samples"           "common.dispersion"
## [4] "pseudo.counts"     "pseudo.lib.size"   "AveLogCPM"
```

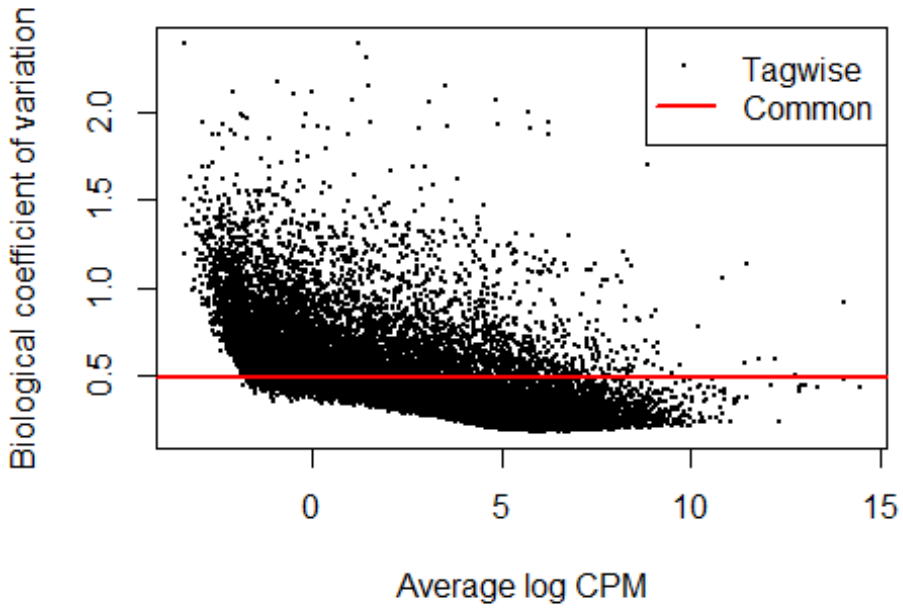
Para el análisis de expresión diferencial, vamos a utilizar dispersiones empíricas de Bayes. Hay que tener en cuenta que es necesario estimar la dispersión común antes de estimar las dispersiones por etiquetas.

```
d1 <- estimateTagwiseDisp(d1)
names(d1)
```

```
##  [1] "counts"             "samples"            "common.dispersion"
##  [4] "pseudo.counts"      "pseudo.lib.size"    "AveLogCPM"
##  [7] "prior.df"           "prior.n"            "tagwise.dispersion"
## [10] "span"
```

La función plotBCV() traza el coeficiente de variación biológica a nivel de etiqueta (raíz cuadrada de dispersiones) frente a log2-CPM.

```
plotBCV(d1)
```




Podemos ver que una sola estimación del coeficiente de variación no es un buen modelo, ya que la dispersión aumenta a medida que aumenta el recuento por millón (CPM).

Ahora calcularemos las estimaciones de dispersión con GLM:

Primero calcularemos la matriz de diseño:

```
designmat <- model.matrix(~ 0 + dgList$samples$group)
designmat
```

```
##    dgList$samples$groupELI dgList$samples$groupNIT dgList$samples$grou
pSFI
## 1                        1                       0
0
```

```
## 2                        1                        0
0
## 3                        1                        0
0
## 4                        1                        0
0
## 5                        1                        0
0
## 6                        1                        0
0
## 7                        1                        0
0
## 8                        1                        0
0
## 9                        1                        0
0
## 10                       1                        0
0
## 11                       0                        1
0
## 12                       0                        1
0
## 13                       0                        1
0
## 14                       0                        1
0
## 15                       0                        1
0
## 16                       0                        1
0
## 17                       0                        1
0
## 18                       0                        1
0
## 19                       0                        1
0
## 20                       0                        1
0
## 21                       0                        0
1
## 22                       0                        0
1
## 23                       0                        0
1
## 24                       0                        0
1
## 25                       0                        0
1
## 26                       0                        0
1
```

```
## 27                          0                          0
1
## 28                          0                          0
1
## 29                          0                          0
1
## 30                          0                          0
1
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$`dgList$samples$group`
## [1] "contr.treatment"

colnames(designmat) <- levels(dgList$samples$group)
```

La dispersión común estima el BCV general del conjunto de datos, promediado sobre todos los genes.

```
d2 <- estimateGLMCommonDisp(dgList,designmat)
```

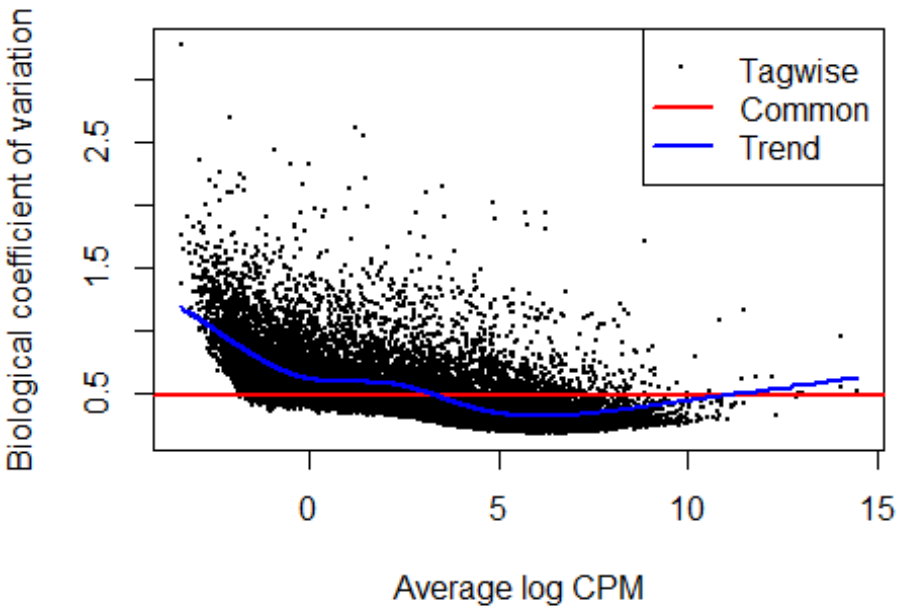
Ahora haremos las estimaciones de dispersión delos genes:

```
d2 <- estimateGLMCommonDisp(dgList,designmat)
d2 <- estimateGLMTrendedDisp(d2,designmat)
# podemos usar el metodo "auto", "bin.spline", "power", "spline", "bin.lo
ess"

d2 <- estimateGLMTagwiseDisp(d2,designmat)
```

Hacemos una gráfica de las dispersiones estimadas:

```
plotBCV(d2)
```




### Comparacion entre los modelos DESeq y edgeR

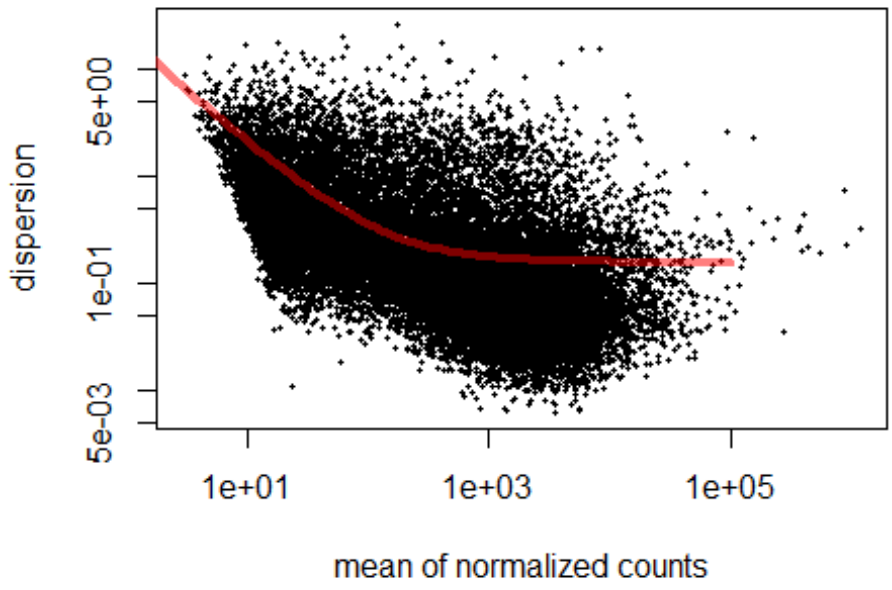
Veamos los resultados usando DESeq:

```
cds <- newCountDataSet(data.frame(dgList$counts), dgList$samples$group)
cds <- estimateSizeFactors(cds)
sizeFactors(cds)

##  GTEX.ZYY3.1926.SM.5GZXS  GTEX.YJ89.0726.SM.5P9F7 GTEX.11XUK.0226.SM.5
EQLW
##                0.9127543                1.5656968                0.941
4113
##  GTEX.YFC4.2626.SM.5P9FQ GTEX.13NZ9.1126.SM.5MR37  GTEX.R55G.0726.SM.2
TC6J
##                1.7190780                1.3531099                0.308
6861
##  GTEX.PLZ4.1226.SM.2I5FE  GTEX.TMMY.0826.SM.33HB9 GTEX.14AS3.0226.SM.5
Q5B6
##                1.2616073                1.4984118                0.892
3937
## GTEX.13QJC.0826.SM.5RQKC  GTEX.QV31.0726.SM.3GAEG GTEX.13OW7.0826.SM.5
L3EL
##                0.9414876                0.9125279                0.823
7525
##  GTEX.X8HC.0726.SM.46MWG GTEX.11DXX.0226.SM.5P9HL  GTEX.Q734.0526.SM.2
I3EH
##                0.9628535                1.4468140                1.015
```

```
7057
## GTEX.13113.0126.SM.5LZVX  GTEX.R3RS.0726.SM.3GIJR GTEX.13S86.1126.SM.5
RQJX
##                0.8509219               0.2428189               0.722
9740
## GTEX.13FTY.0726.SM.5J2OH  GTEX.ZYFC.0926.SM.5GZWW  GTEX.QLQ7.0726.SM.2
I5G2
##                1.1876266               0.9899802               1.485
4682
##  GTEX.ZLV1.0126.SM.4WWBZ  GTEX.Y5V6.0526.SM.4VBRV GTEX.13FH7.0126.SM.5
KLZ1
##                1.0337837               1.2211305               1.213
5299
## GTEX.13NZ8.0226.SM.5J2OK  GTEX.R55C.0626.SM.2TF4Q  GTEX.WYVS.0326.SM.3
NM9V
##                1.1958195               0.7654034               1.529
1694
## GTEX.131YS.0726.SM.5P9G9 GTEX.11GS4.0826.SM.5986J GTEX.13FXS.0726.SM.5
LZXJ
##                1.3806445               0.9543062               0.953
2746
```

```r
cds <- estimateDispersions( cds , method="blind")
plotDispEsts(cds)
```




En este gráfico se traza la dispersión en el eje vertical en lugar del coeficiente de variación biológica.

### Expresión diferencial

Una vez que se estiman las dispersiones, podemos proceder con los procedimientos de prueba para determinar la expresión diferencial. La función exactTest()lleva a cabo pruebas con etiquetas usando la prueba binomial negativa exacta. La topTags()función muestra los resultados de las pruebas para las n etiquetas más significativas . Por defecto, el algoritmo de Benjamini y Hochberg se usa para controlar los FDR.

Primero lo haremos para d1 en el que solo habia una dispersión comun:

```
et12 <- exactTest(d1, pair=c(1,2)) # compara grupos 1 y 2
et13 <- exactTest(d1, pair=c(1,3)) # compara grupos 1 y 3
et23 <- exactTest(d1, pair=c(2,3)) # compara grupos 2 y 3

topTags(et12)
```

```
## Comparison of groups:  NIT-ELI
##                     logFC   logCPM       PValue          FDR
## ENSG00000105369 -7.959291 5.370352 1.503462e-18 1.656635e-14
## ENSG00000083454 -6.724115 4.610802 1.586068e-18 1.656635e-14
## ENSG00000143297 -7.039446 5.235187 2.258843e-18 1.656635e-14
## ENSG00000136573 -7.100350 4.633171 7.348041e-18 4.041790e-14
## ENSG00000035720 -6.931713 1.704583 2.991016e-17 1.316167e-13
## ENSG00000132704 -8.095621 3.776334 6.126310e-17 2.246518e-13
## ENSG00000211893 -7.432055 8.143807 8.949981e-17 2.813107e-13
## ENSG00000132465 -6.159953 7.569634 2.258388e-16 6.211131e-13
## ENSG00000110777 -6.075717 5.478448 3.502111e-16 7.728291e-13
## ENSG00000174123 -6.452497 3.312477 3.578128e-16 7.728291e-13
```

```
topTags(et13)
```

```
## Comparison of groups:  SFI-ELI
##                      logFC     logCPM       PValue          FDR
## ENSG00000152952  1.1579954  6.5928899 5.499935e-09 0.0001210096
## ENSG00000114270 -2.1261371  4.0917445 3.601060e-08 0.0003961526
## ENSG00000164638  1.8385594  4.4874509 9.217038e-08 0.0006139220
## ENSG00000246575 -1.5595843  0.8116396 1.116120e-07 0.0006139220
## ENSG00000230937 -3.3454314  0.1182087 1.864241e-07 0.0007186033
## ENSG00000235111 -1.9815532  0.6971040 1.959649e-07 0.0007186033
## ENSG00000117450  1.0882947  8.6060029 3.322718e-07 0.0010443776
## ENSG00000254029 -6.2054585 -1.7146513 4.153228e-07 0.0011422416
## ENSG00000164023  1.1871192  5.0832377 5.131068e-07 0.0011528167
## ENSG00000091164  0.8840216  7.0897588 5.239600e-07 0.0011528167
```

```
topTags(et23)
```

```
## Comparison of groups:  SFI-NIT
##                     logFC   logCPM       PValue          FDR
## ENSG00000132465 5.790584 7.569634 4.212307e-15 9.267917e-11
## ENSG00000211900 6.861673 3.418219 2.019553e-14 1.929256e-10
```

```
## ENSG00000105369 6.422645 5.370352 3.410472e-14 1.929256e-10
## ENSG00000211966 7.148734 4.607411 3.507420e-14 1.929256e-10
## ENSG00000211598 6.961605 6.072348 5.262867e-14 2.315872e-10
## ENSG00000211947 6.780802 4.162516 8.061528e-14 2.956162e-10
## ENSG00000240041 6.436346 3.953837 1.097040e-13 3.190175e-10
## ENSG00000211935 7.793636 3.096525 1.180827e-13 3.190175e-10
## ENSG00000242887 6.820799 2.397691 1.306763e-13 3.190175e-10
## ENSG00000241351 6.807645 5.966672 1.449948e-13 3.190175e-10
```

El número total de genes expresados diferencialmente en FDR <0.05 es:

```
de12 <- decideTestsDGE(et12, adjust.method="BH", p.value=0.05)
de13 <- decideTestsDGE(et13, adjust.method="BH", p.value=0.05)
de23 <- decideTestsDGE(et23, adjust.method="BH", p.value=0.05)
summary(de12)
```

```
##        NIT-ELI
## Down      2109
## NotSig   19051
## Up         842
```

```
summary(de13)
```

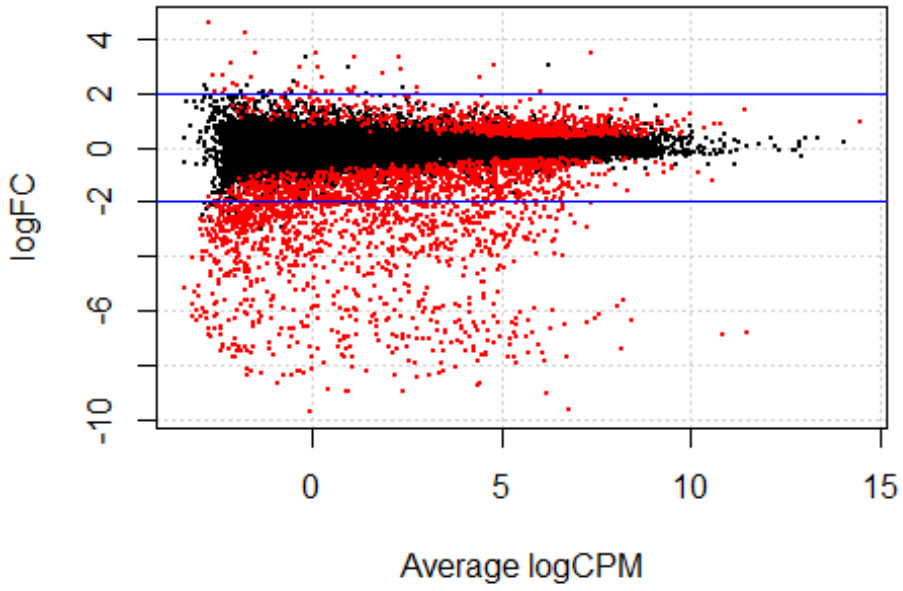
```
##        SFI-ELI
## Down       775
## NotSig   20580
## Up         647
```

```
summary(de23)
```

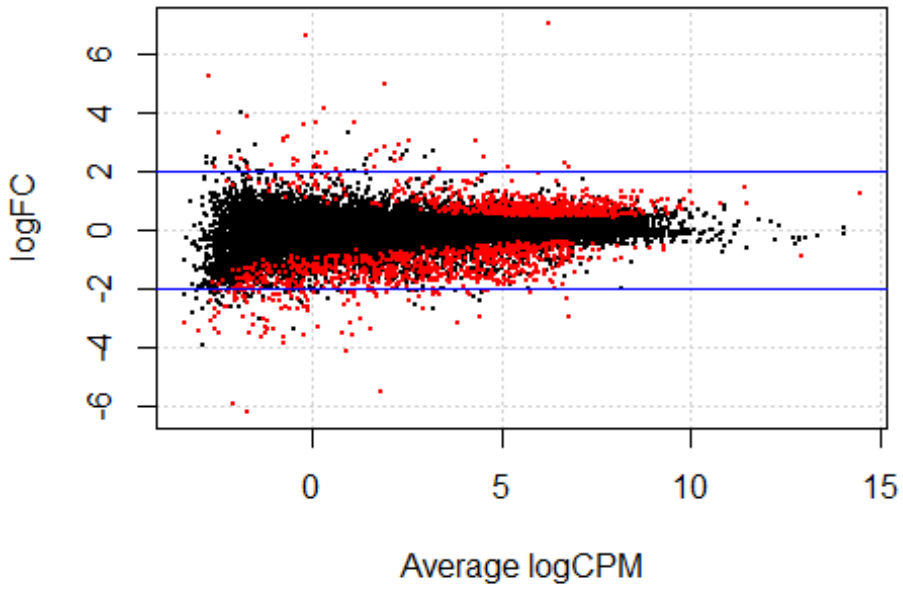
```
##        SFI-NIT
## Down        30
## NotSig   21411
## Up         561
```

Se nos muestran las etiquetas infraexpresadas, no expresadas diferencialmente y sobreexpresadas, respectivamente.

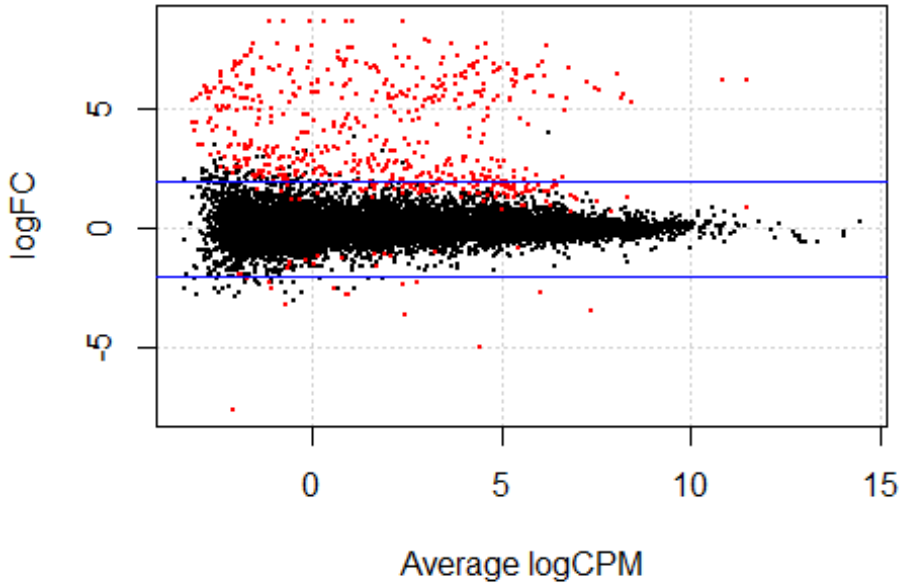
```
de12tags12 <- rownames(d1)[as.logical(de12)]
de13tags13 <- rownames(d1)[as.logical(de13)]
de23tags23 <- rownames(d1)[as.logical(de23)]
plotSmear(et12, de.tags=de12tags12)
abline(h = c(-2, 2), col = "blue")
```




```
plotSmear(et13, de.tags=de13tags13)
abline(h = c(-2, 2), col = "blue")
```



```
plotSmear(et23, de.tags=de23tags23)
abline(h = c(-2, 2), col = "blue")
```




Ahora haremos la expresion diferencial con GLM (d2).

Ajustamos el modelo lineal

```
fit <- glmFit(d2, designmat)
names(fit)
```

```
##  [1] "coefficients"          "fitted.values"         "deviance"
##  [4] "method"                "counts"                "unshrunk.coeffic
ients"
##  [7] "df.residual"           "design"                "offset"
## [10] "dispersion"            "prior.count"           "samples"
## [13] "prior.df"              "AveLogCPM"
```

```
head(coef(fit))
```

```
##                        ELI       NIT       SFI
## ENSG00000227232 -11.17795 -11.10555 -11.18483
## ENSG00000233750 -13.54019 -14.29986 -14.58753
## ENSG00000237683 -11.13330 -10.82992 -11.36526
## ENSG00000239906 -15.40420 -14.93073 -15.53459
## ENSG00000241860 -13.17183 -13.16994 -13.56915
## ENSG00000228463 -14.04140 -14.01043 -13.81948
```

Realizamos las pruebas y le decimos que muestre los genes principales:

```
lrt12 <- glmLRT(fit, contrast=c(1,-1,0))
lrt13 <- glmLRT(fit, contrast=c(1,0,-1))
lrt23 <- glmLRT(fit, contrast=c(0,1,-1))
topTags(lrt12)
```

```
## Coefficient:  1*ELI -1*NIT
##                     logFC   logCPM       LR       PValue          FDR
## ENSG00000083454 6.724404 4.610790 78.06307 9.980274e-19 8.075012e-15
## ENSG00000105369 7.959764 5.370352 78.01250 1.023908e-18 8.075012e-15
## ENSG00000143297 7.039684 5.235190 77.86904 1.101038e-18 8.075012e-15
## ENSG00000136573 7.100830 4.633165 74.48283 6.116809e-18 3.364551e-14
## ENSG00000035720 6.934206 1.704591 71.63180 2.593436e-17 1.141215e-13
## ENSG00000132704 8.096907 3.776319 70.91064 3.737766e-17 1.370639e-13
## ENSG00000211893 7.432092 8.143808 69.91506 6.191385e-17 1.946041e-13
## ENSG00000132465 6.159986 7.569635 68.39030 1.341358e-16 3.689069e-13
## ENSG00000110777 6.075844 5.478449 67.92249 1.700505e-16 4.157169e-13
## ENSG00000174123 6.453040 3.312501 67.50060 2.106227e-16 4.634120e-13
```

```
topTags(lrt13)
```

```
## Coefficient:  1*ELI -1*SFI
##                      logFC    logCPM       LR       PValue          FD
R
## ENSG00000152952 -1.1579906 6.5928896 34.53262 4.191668e-09 9.222508e-0
5
## ENSG00000114270  2.1261204 4.0917531 31.39518 2.105025e-08 2.315738e-0
4
## ENSG00000164638 -1.8385325 4.4874560 29.09864 6.878522e-08 5.044708e-0
4
## ENSG00000246575  1.5594593 0.8116321 28.03271 1.192823e-07 5.559205e-0
4
## ENSG00000230937  3.3445600 0.1184788 27.92156 1.263341e-07 5.559205e-0
4
## ENSG00000235111  1.9813407 0.6971852 27.30972 1.733381e-07 6.356308e-0
4
## ENSG00000164023 -1.1871107 5.0832415 25.63372 4.127637e-07 1.084353e-0
3
## ENSG00000091164 -0.8840174 7.0897583 25.50468 4.413103e-07 1.084353e-0
3
## ENSG00000065833 -1.6544754 4.0982129 25.38601 4.693100e-07 1.084353e-0
3
## ENSG00000011201 -0.9262465 4.3598616 25.28759 4.938759e-07 1.084353e-0
3
```

```
topTags(lrt23)
```

```
## Coefficient:  1*NIT -1*SFI
##                     logFC   logCPM       LR       PValue          FDR
## ENSG00000132465 -5.790607 7.569635 62.55978 2.584785e-15 5.687044e-11
## ENSG00000211900 -6.862566 3.418238 59.51218 1.215397e-14 1.337058e-10
## ENSG00000105369 -6.422998 5.370352 58.12292 2.462403e-14 1.728339e-10
```

```
## ENSG00000211966 -7.149144 4.607428 57.41076 3.536664e-14 1.728339e-10
## ENSG00000254395 -8.444301 1.032817 57.20451 3.927686e-14 1.728339e-10
## ENSG00000211598 -6.961892 6.072352 56.70652 5.059563e-14 1.786294e-10
## ENSG00000211947 -6.780567 4.162529 56.21467 6.497487e-14 1.786294e-10
## ENSG00000240041 -6.436744 3.953852 56.14523 6.731065e-14 1.786294e-10
## ENSG00000211935 -7.795932 3.096583 55.98384 7.306903e-14 1.786294e-10
## ENSG00000241351 -6.807787 5.966674 55.60578 8.856266e-14 1.948556e-10
```

El número total de genes expresados diferencialmente en FDR <0.05 es:

```
de2en <- decideTestsDGE(lrt12, adjust.method="BH", p.value = 0.05)
de2es <- decideTestsDGE(lrt13, adjust.method="BH", p.value = 0.05)
de2ns <- decideTestsDGE(lrt23, adjust.method="BH", p.value = 0.05)
de2tagsen <- rownames(d2)[as.logical(de2en)]
de2tagses <- rownames(d2)[as.logical(de2es)]
de2tagsns <- rownames(d2)[as.logical(de2ns)]
summary(de2en)
```

```
##        1*ELI -1*NIT
## Down             876
## NotSig         18999
## Up              2127
```

```
summary(de2es)
```

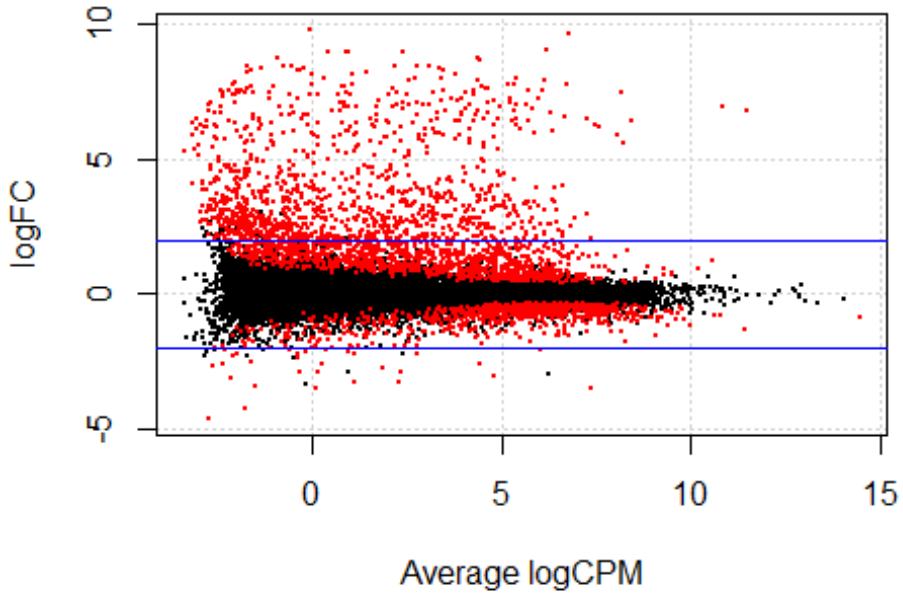
```
##        1*ELI -1*SFI
## Down             674
## NotSig         20521
## Up               807
```

```
summary(de2ns)
```

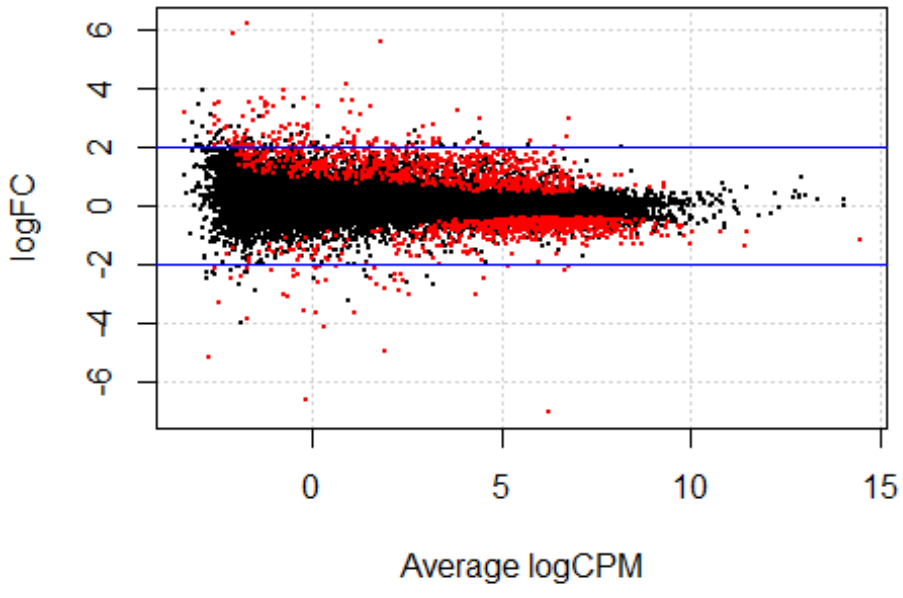
```
##        1*NIT -1*SFI
## Down             561
## NotSig         21410
## Up                31
```

Veamos ahora los graficos para cada contraste:

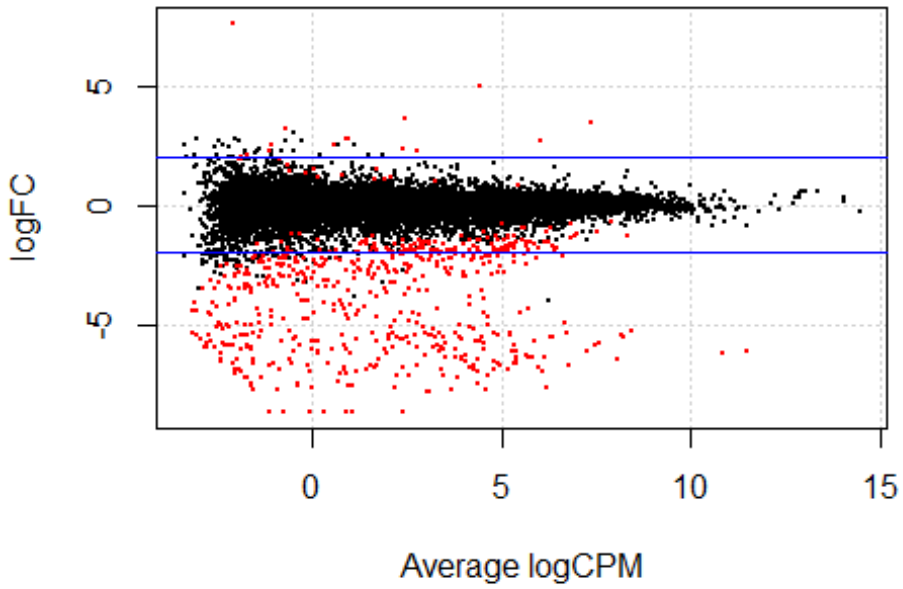
```
plotSmear(lrt12, de.tags=de2tagsen)
abline(h = c(-2, 2), col = "blue")
```




```
plotSmear(lrt13, de.tags=de2tagses)
abline(h = c(-2, 2), col = "blue")
```



```
plotSmear(lrt23, de.tags=de2tagsns)
abline(h = c(-2, 2), col = "blue")
```




```
results <- as.data.frame(topTags(lrt12,n = Inf))
head(results)
```

```
##                     logFC   logCPM       LR       PValue          FDR
## ENSG00000083454 6.724404 4.610790 78.06307 9.980274e-19 8.075012e-15
## ENSG00000105369 7.959764 5.370352 78.01250 1.023908e-18 8.075012e-15
## ENSG00000143297 7.039684 5.235190 77.86904 1.101038e-18 8.075012e-15
## ENSG00000136573 7.100830 4.633165 74.48283 6.116809e-18 3.364551e-14
## ENSG00000035720 6.934206 1.704591 71.63180 2.593436e-17 1.141215e-13
## ENSG00000132704 8.096907 3.776319 70.91064 3.737766e-17 1.370639e-13
```

```
dim(results)
```

```
## [1] 22002     5
```

```
results2 <- as.data.frame(topTags(lrt13,n = Inf))
head(results2)
```

```
##                     logFC    logCPM       LR       PValue          FDR
## ENSG00000152952 -1.157991 6.5928896 34.53262 4.191668e-09 9.222508e-05
## ENSG00000114270  2.126120 4.0917531 31.39518 2.105025e-08 2.315738e-04
## ENSG00000164638 -1.838532 4.4874560 29.09864 6.878522e-08 5.044708e-04
## ENSG00000246575  1.559459 0.8116321 28.03271 1.192823e-07 5.559205e-04
## ENSG00000230937  3.344560 0.1184788 27.92156 1.263341e-07 5.559205e-04
## ENSG00000235111  1.981341 0.6971852 27.30972 1.733381e-07 6.356308e-04
```

```
dim(results2)
```

```
## [1] 22002     5
```

```
results3 <- as.data.frame(topTags(lrt23,n = Inf))
head(results3)
```

```
##                     logFC   logCPM       LR       PValue          FDR
## ENSG00000132465 -5.790607 7.569635 62.55978 2.584785e-15 5.687044e-11
## ENSG00000211900 -6.862566 3.418238 59.51218 1.215397e-14 1.337058e-10
## ENSG00000105369 -6.422998 5.370352 58.12292 2.462403e-14 1.728339e-10
## ENSG00000211966 -7.149144 4.607428 57.41076 3.536664e-14 1.728339e-10
## ENSG00000254395 -8.444301 1.032817 57.20451 3.927686e-14 1.728339e-10
## ENSG00000211598 -6.961892 6.072352 56.70652 5.059563e-14 1.786294e-10
```

```
dim(results3)
```

```
## [1] 22002     5
```

```
summary(de <- decideTestsDGE(lrt12,
                             adjust.method="BH", p.value = 0.05))
```

```
##        1*ELI -1*NIT
## Down             876
## NotSig         18999
## Up              2127
```

```
save(lrt12,
     lrt13,
     lrt23,
     dgList,grupos,file="C:/Pec2DatosOmicos/results/DE.Rdata")
```

### Anotación y visualización de resultados

Para anotar nuestros resultados, vamos a quedarnos con los símbolos genéticos y el nombre completo del gen. Separaremos la información de anotación en un marco de datos usando la funcion select.

Ajunto el codigo de results2 y results3 en el apendice.

```
ann <- select(org.Hs.eg.db,keys=rownames(results), keytype = "ENSEMBL",
              columns=c("SYMBOL","GENENAME"))
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
head(ann)
```

```
##           ENSEMBL SYMBOL                                        GENENAM
E
## 1 ENSG00000083454  P2RX5                      purinergic receptor P2X
5
## 2 ENSG00000105369  CD79A                                CD79a molecul
e
```

```
## 3 ENSG00000143297  FCRL5                              Fc receptor like
5
## 4 ENSG00000136573    BLK BLK proto-oncogene, Src family tyrosine kinas
e
## 5 ENSG00000035720  STAP1     signal transducing adaptor family member
1
## 6 ENSG00000132704  FCRL2                              Fc receptor like
2
```

```
dim(ann)
```

```
## [1] 22143     3
```

```
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
```

Verifiquemos nuevamente que la columna ENSEMBL coincida exactamente con los nombres de las filas de results.

```
table(unique(ann$ENSEMBL)==rownames(results))
```

```
##
##  TRUE
## 22002
```

```
# Tengo que hacer esto debido a la salida 'select()' returned 1:many...
ann <- ann[!duplicated(ann$ENSEMBL), ]
results.annotated <- cbind(results, ann)
```

```
head(results.annotated)
```

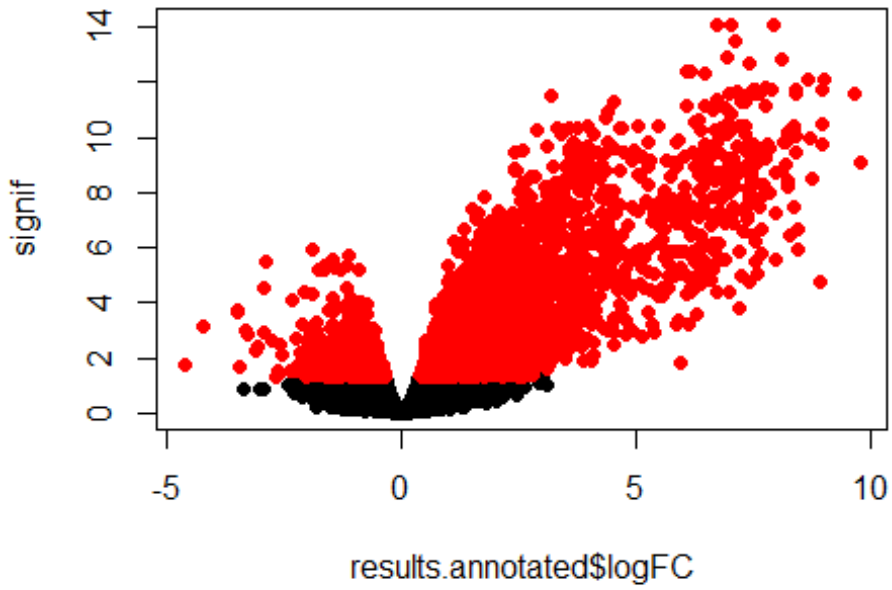
```
##                     logFC   logCPM       LR       PValue          FDR
## ENSG00000083454 6.724404 4.610790 78.06307 9.980274e-19 8.075012e-15
## ENSG00000105369 7.959764 5.370352 78.01250 1.023908e-18 8.075012e-15
## ENSG00000143297 7.039684 5.235190 77.86904 1.101038e-18 8.075012e-15
## ENSG00000136573 7.100830 4.633165 74.48283 6.116809e-18 3.364551e-14
## ENSG00000035720 6.934206 1.704591 71.63180 2.593436e-17 1.141215e-13
## ENSG00000132704 8.096907 3.776319 70.91064 3.737766e-17 1.370639e-13
##                          ENSEMBL SYMBOL
## ENSG00000083454 ENSG00000083454  P2RX5
## ENSG00000105369 ENSG00000105369  CD79A
## ENSG00000143297 ENSG00000143297  FCRL5
## ENSG00000136573 ENSG00000136573    BLK
## ENSG00000035720 ENSG00000035720  STAP1
## ENSG00000132704 ENSG00000132704  FCRL2
##                                                    GENENAME
## ENSG00000083454                      purinergic receptor P2X 5
## ENSG00000105369                                CD79a molecule
## ENSG00000143297                           Fc receptor like 5
## ENSG00000136573 BLK proto-oncogene, Src family tyrosine kinase
## ENSG00000035720     signal transducing adaptor family member 1
## ENSG00000132704                           Fc receptor like 2
```

```
write.csv(results.annotated,file="C:/Pec2DatosOmicos/results/ELIVsNIT.csv
",
          row.names=FALSE)
```

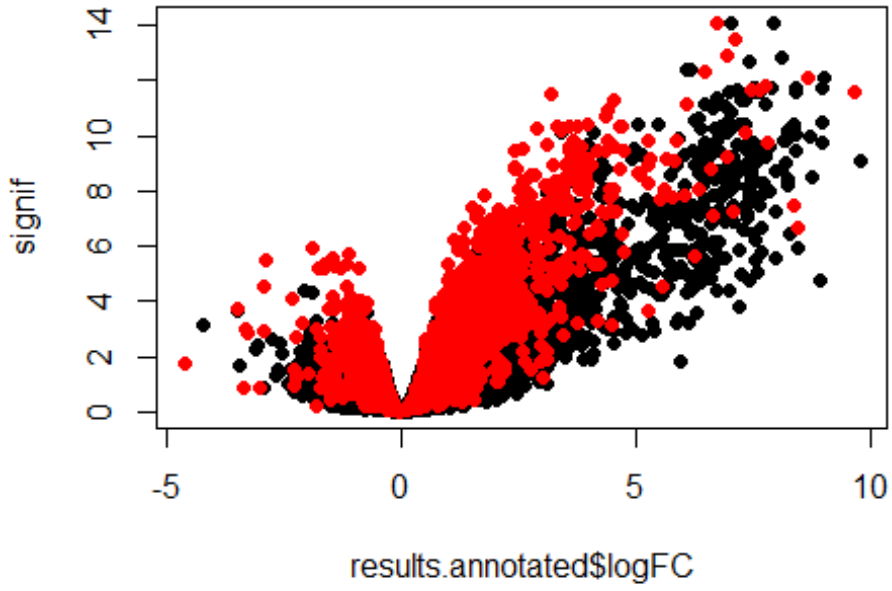
Una alternativa es utilizar BioMart . BioMart es mucho más completo, pero los "organism packages" se ajustan mejor al flujo de trabajo de Bioconductor.

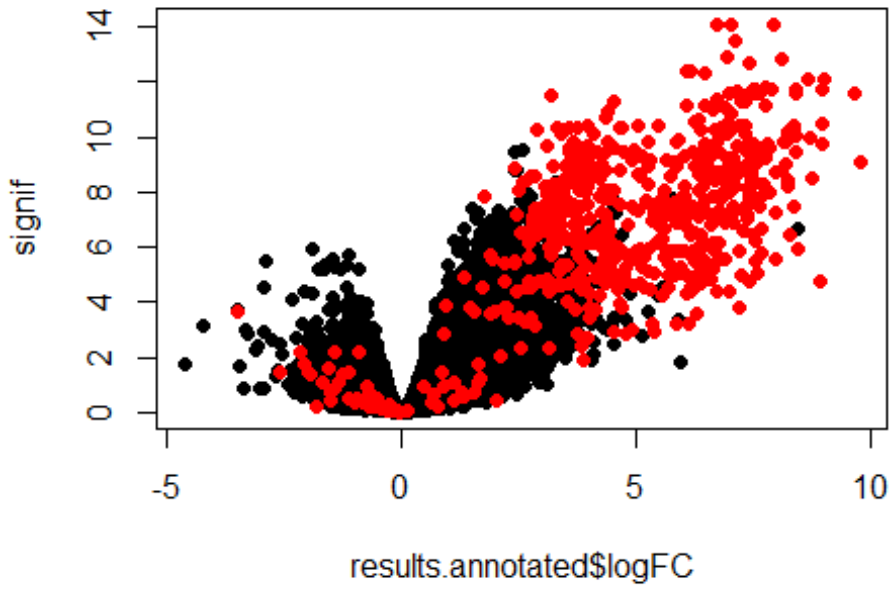
Veamos también como queda representado con un VolcanoPlot:

```
detags <- rownames(dgList)[as.logical(de)]
signif <- -log10(results.annotated$FDR)
plot(results.annotated$logFC,signif,pch=16)
points(results.annotated[detags,"logFC"],-log10(results.annotated[detags,
"FDR"]),pch=16,col="red")
```




```
#ggplot(results, aes(x = logFC, y=-log10(FDR))) + geom_point()
```




Del mismo modo que hicimos anteriormente, podemos ver graficos interctivos con el paquete Glima (lo hago solo para la EvsN):

```
normCounts <- dgList$counts
glXYPlot(x=results$logFC, y=-log10(results$FDR),
         xlab="logFC", ylab="B", main="EVsN",
         counts=normCounts, groups=grupos, status=de,
         id.column="ENSEMBL", folder="volcano")
```

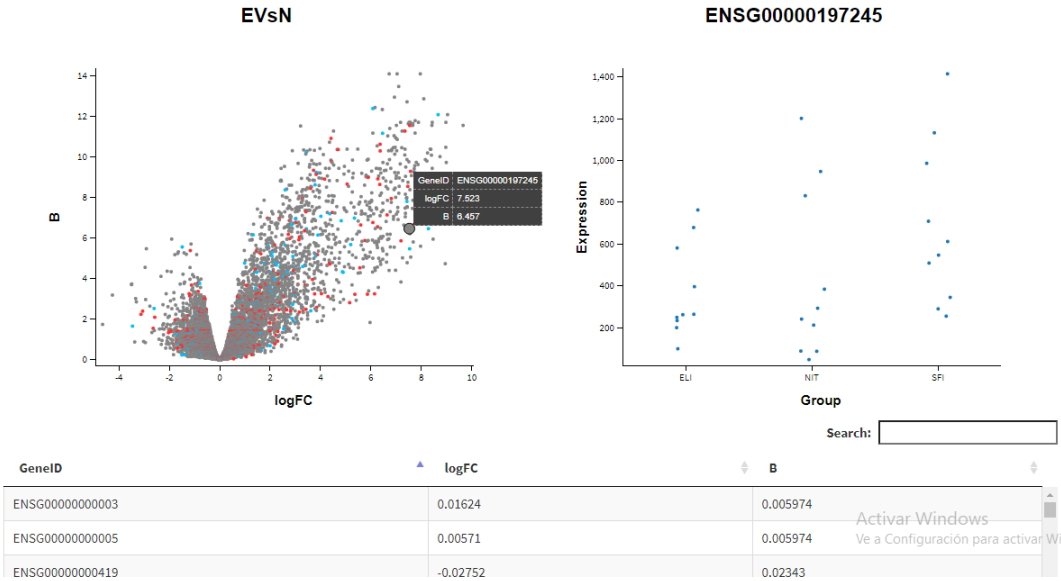



*Diagrama MDS interactivo*

Se podrian hacer más cosas, como recuperar las ubicaciones genomicas, manipular los intervalos genomicos con GenomicRangers, exportar pistas o extraer lecturas.

## Significación biologica

GOseq es un método para realizar análisis de ontología génica (GO) adecuado para datos de RNA-seq, ya que explica el sesgo de la longitud del gen en la detección de sobrerepresentación.

```
# lista de DEGs filtrando con FDR
genes <- results$FDR < 0.05



# Añadimos nombres
names(genes) <- rownames(results)
print(head(genes))

## ENSG00000083454 ENSG00000105369 ENSG00000143297 ENSG00000136573 ENSG00
000035720
##            TRUE            TRUE            TRUE            TRUE
TRUE
## ENSG00000132704
##            TRUE
```

Calcularemos una función de ponderación de probabilidad o PWF que puede considerarse como una función que da la probabilidad de que un gen se exprese diferencialmente (DE), basándose solo en su longitud.

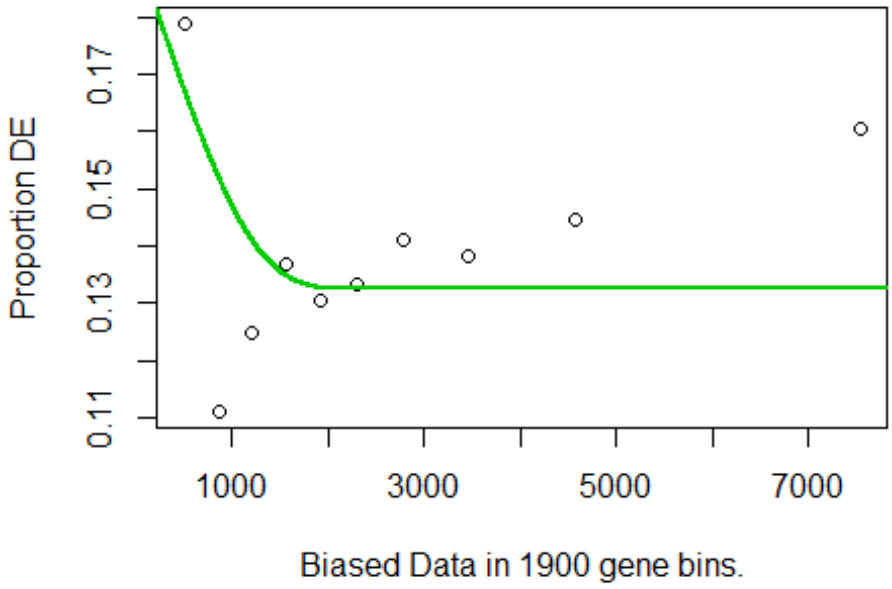
```
supportedOrganisms()[supportedOrganisms()$Genome=="hg19",]
```

```
## Loading required package: rtracklayer
```

```
##    Genome         Id  Id Description Lengths in geneLeneDataBase
## 4    hg19  knownGene  Entrez Gene ID                         TRUE
## 36   hg19    ensGene Ensembl gene ID                         TRUE
## 81   hg19 geneSymbol     Gene Symbol                         TRUE
##    GO Annotation Available
## 4                     TRUE
## 36                    TRUE
## 81                    TRUE
```

```
pwf <- nullp(genes, "hg19", "ensGene")
```

```
## Loading hg19 length data...
```



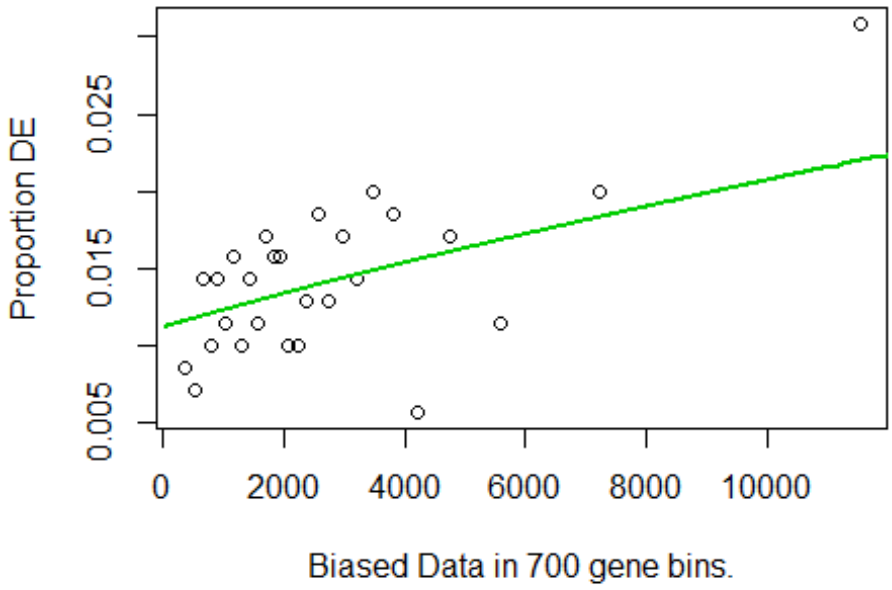

```
head(pwf)
```

```
##                 DEgenes bias.data       pwf
## ENSG00000083454    TRUE    1520.0 0.1353301
## ENSG00000105369    TRUE    1206.0 0.1411115
## ENSG00000143297    TRUE    2196.5 0.1324122
## ENSG00000136573    TRUE    2483.5 0.1324118
```

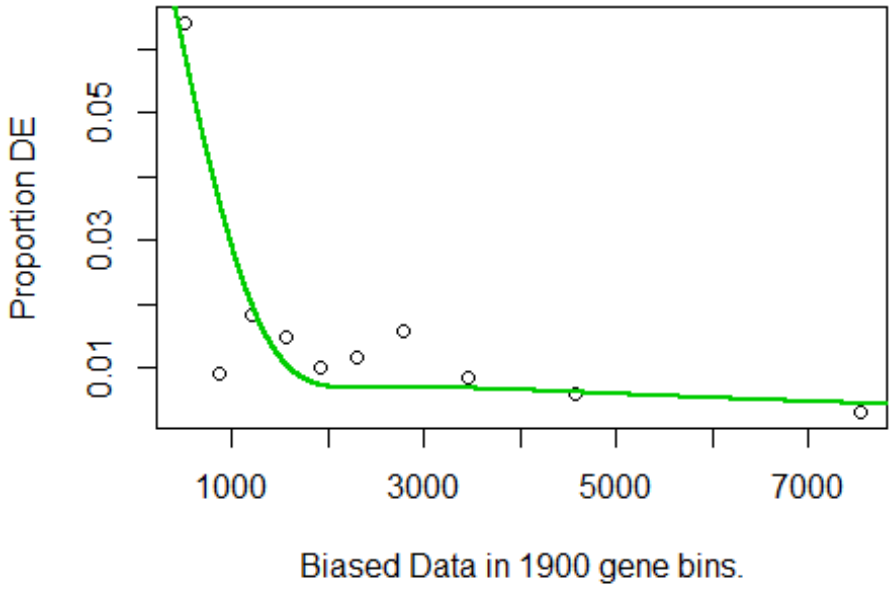
```
## ENSG00000035720    TRUE    1338.0 0.1381760
## ENSG00000132704    TRUE    2591.0 0.1324118

## Loading hg19 length data...

## Warning in pcls(G): initial point very close to some inequality constr
aints
```




```
## Loading hg19 length data...
```




```
write.csv(results.annotated,file="C:/Pec2DatosOmicos/results/pwf.tsv",
          row.names=FALSE)
```

Las gráficas salen diferente a todas las que he visto en diferentes documentos, no se si es debido a algun fallo en el analisis, o que el ajuste es malo.

He probado a hacerlo de esta otra forma, y se obtiene el mismo resultado:

```
geness =as.integer(p.adjust(et12$table$PValue[et12$table$logFC!=0],
                                method="BH")<.05)
                                names(geness)=row.names(et12$table[et12$ta
ble$logFC!=0,])
pwff <- nullp(geness, "hg19", "ensGene")
```

Realizamos un análisis de enriquecimiento del conjunto de genes:

```
go.results <- goseq(pwf, "hg19", "ensGene")

## Fetching GO annotations...

## For 6343 genes, we could not find any categories. These genes will be
excluded.

## To force their use, please run with use_genes_without_cat=TRUE (see do
cumentation).

## This was the default behavior for version 1.15.1 and earlier.

## Calculating the p-values...
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
head(go.results)
```

```
##         category over_represented_pvalue under_represented_pvalue numD
EInCat
## 1024  GO:0002376           1.127899e-58                         1
621
## 2816  GO:0005886           4.298089e-53                         1
875
## 16713 GO:0071944           1.006569e-52                         1
891
## 3517  GO:0006955           5.775350e-51                         1
458
## 12622 GO:0046649           1.705570e-48                         1
220
## 938   GO:0002250           8.795023e-48                         1
162
##       numInCat                      term ontology
## 1024      2582    immune system process       BP
## 2816      4298          plasma membrane       CC
## 16713     4410           cell periphery       CC
## 3517      1762          immune response       BP
## 12622      604    lymphocyte activation       BP
## 938        370 adaptive immune response       BP
```

```
enriched.GO=go.results$category[p.adjust(go.results$over_represented_pval
ue,
                                         method="BH")<.05]
```

```
head(enriched.GO)
```

```
## [1] "GO:0002376" "GO:0005886" "GO:0071944" "GO:0006955" "GO:0046649"
## [6] "GO:0002250"
```

Categorías GO relacionadas:

```
for(go in enriched.GO[1:5]){
  print(GOTERM[[go]])
  cat("--------------------------------------\n")
  }
```

```
## GOID: GO:0002376
## Term: immune system process
## Ontology: BP
## Definition: Any process involved in the development or functioning of
##     the immune system, an organismal system for calibrated responses t
o
##     potential internal or invasive threats.
## --------------------------------------
## GOID: GO:0005886
```

```
## Term: plasma membrane
## Ontology: CC
## Definition: The membrane surrounding a cell that separates the cell
##     from its external environment. It consists of a phospholipid
##     bilayer and associated proteins.
## Synonym: bacterial inner membrane
## Synonym: cell membrane
## Synonym: cellular membrane
## Synonym: cytoplasmic membrane
## Synonym: inner endospore membrane
## Synonym: juxtamembrane
## Synonym: plasma membrane lipid bilayer
## Synonym: plasmalemma
## Synonym: GO:0005904
## Secondary: GO:0005904
## -------------------------------------
## GOID: GO:0071944
## Term: cell periphery
## Ontology: CC
## Definition: The part of a cell encompassing the cell cortex, the plasm
a
##     membrane, and any external encapsulating structures.
## -------------------------------------
## GOID: GO:0006955
## Term: immune response
## Ontology: BP
## Definition: Any immune system process that functions in the calibrated
##     response of an organism to a potential internal or invasive threat
.
## -------------------------------------
## GOID: GO:0046649
## Term: lymphocyte activation
## Ontology: BP
## Definition: A change in morphology and behavior of a lymphocyte
##     resulting from exposure to a specific antigen, mitogen, cytokine,
##     chemokine, cellular ligand, or soluble factor.
## -------------------------------------
```

También podriamos haber hecho el analisis de significación biologica con la herramienta en linea Enrich, para lo cual necesitariamos subir a la plataforma de Enrich el archivo con las anotaciones de los genes.

El paquete fgsea que aleatoriza reiteradamente las etiquetas de las muestras y vuelve a realizar pruebas de enriquecimiento en las clases aleatorias.

## Resultados

Aqui se mostrara una lista de archivos generados en el estudio de caso actual.

```
listOfFiles <- dir("./results/")
knitr::kable(
  listOfFiles, booktabs = TRUE,
  caption = 'List of files generated in the analysis',
  col.names="List_of_Files"
)
```

List of files generated in the analysis
List_of_Files
DE.Rdata
ELIVsNIT.csv
ELIVsSFI.csv
Glima1.png
Glima2.png
NITVsSFI.csv
preprocessing.Rdata
pwf.tsv
pwf2.tsv
pwf3.tsv

# Apendice

## Anotación y visualización de resultados

```
ann2 <- select(org.Hs.eg.db,keys=rownames(results2), keytype = "ENSEMBL",
               columns=c("SYMBOL","GENENAME"))
ann3 <- select(org.Hs.eg.db,keys=rownames(results3), keytype = "ENSEMBL",
               columns=c("SYMBOL","GENENAME"))
```

Verifiquemos nuevamente que la columna ENSEMBL coincida exactamente con los nombres de las filas de results.

```
# Tengo que hacer esto debido a la salida 'select()' returned 1:many...
ann2 <- ann2[!duplicated(ann2$ENSEMBL), ]
results.annotated2 <- cbind(results2, ann2)



ann3 <- ann3[!duplicated(ann3$ENSEMBL), ]
results.annotated3 <- cbind(results3, ann3)

detags <- rownames(dgList)[as.logical(de2es)]
signif <- -log10(results.annotated$FDR)
plot(results.annotated$logFC,signif,pch=16)
points(results.annotated[detags,"logFC"],-log10(results.annotated[detags,
"FDR"]),pch=16,col="red")
```

```
#ggplot(results, aes(x = logFC, y=-log10(FDR))) + geom_point()

detags <- rownames(dgList)[as.logical(de2ns)]
signif <- -log10(results.annotated$FDR)
plot(results.annotated$logFC,signif,pch=16)
points(results.annotated[detags,"logFC"],-log10(results.annotated[detags,
"FDR"]),pch=16,col="red")

#ggplot(results, aes(x = logFC, y=-log10(FDR))) + geom_point()
```

## Significación biologica

```
# lista de DEGs filtrando con FDR
genes2 <- results2$FDR < 0.01

# Añadimos nombres
names(genes2) <- rownames(results2)

print(head(genes2))

# lista de DEGs filtrando con FDR
genes3 <- results3$FDR < 0.01

# Añadimos nombres
names(genes3) <- rownames(results3)

print(head(genes3))

pwf2 <- nullp(genes2, "hg19", "ensGene")
head(pwf2)

pwf3 <- nullp(genes3, "hg19", "ensGene")
head(pwf3)

write.csv(results.annotated,file="C:/Pec2DatosOmicos/results/pwf2.tsv",
          row.names=FALSE)
write.csv(results.annotated,file="C:/Pec2DatosOmicos/results/pwf3.tsv",
          row.names=FALSE)

go.results2 <- goseq(pwf2, "hg19", "ensGene")

## Fetching GO annotations...

## For 6343 genes, we could not find any categories. These genes will be
excluded.

## To force their use, please run with use_genes_without_cat=TRUE (see do
cumentation).

## This was the default behavior for version 1.15.1 and earlier.

## Calculating the p-values...
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
head(go.results2)
```

```
##         category over_represented_pvalue under_represented_pvalue numD
EInCat
## 2654  GO:0005615            1.215104e-07                1.0000000
72
## 11674 GO:0044421            3.199790e-07                1.0000000
74
## 13314 GO:0048870            3.309123e-07                0.9999999
47
## 14060 GO:0051674            3.309123e-07                0.9999999
47
## 5874  GO:0016477            4.173663e-07                0.9999999
44
## 7535  GO:0030855            4.203663e-07                0.9999999
26
##       numInCat                              term ontology
## 2654      2703             extracellular space       CC
## 11674     2872       extracellular region part       CC
## 13314     1412                   cell motility       BP
## 14060     1412            localization of cell       BP
## 5874      1283                 cell migration       BP
## 7535       551 epithelial cell differentiation       BP
```

```
enriched.GO2=go.results2$category[p.adjust(go.results2$over_represented_p
value,
                                   method="BH")<.05]
```

```
head(enriched.GO2)
```

```
## [1] "GO:0005615" "GO:0044421" "GO:0048870" "GO:0051674" "GO:0016477"
## [6] "GO:0030855"
```

Categorías GO relacionadas:

```
for(go in enriched.GO2[1:5]){
  print(GOTERM[[go]])
  cat("--------------------------------------\n")
  }
```

```
## GOID: GO:0005615
## Term: extracellular space
## Ontology: CC
## Definition: That part of a multicellular organism outside the cells
##     proper, usually taken to be outside the plasma membranes, and
##     occupied by fluid.
## Synonym: intercellular space
## --------------------------------------
## GOID: GO:0044421
```

```
## Term: extracellular region part
## Ontology: CC
## Definition: Any constituent part of the extracellular region, the spac
e
##     external to the outermost structure of a cell. For cells without
##     external protective or external encapsulating structures this
##     refers to space outside of the plasma membrane. This term covers
##     constituent parts of the host cell environment outside an
##     intracellular parasite.
## Synonym: extracellular structure
## -------------------------------------
## GOID: GO:0048870
## Term: cell motility
## Ontology: BP
## Definition: Any process involved in the controlled self-propelled
##     movement of a cell that results in translocation of the cell from
##     one place to another.
## Synonym: cell locomotion
## Synonym: cell movement
## Synonym: movement of a cell
## -------------------------------------
## GOID: GO:0051674
## Term: localization of cell
## Ontology: BP
## Definition: Any process in which a cell is transported to, and/or
##     maintained in, a specific location.
## Synonym: cell localization
## Synonym: establishment and maintenance of cell localization
## Synonym: establishment and maintenance of localization of cell
## Synonym: localisation of cell
## -------------------------------------
## GOID: GO:0016477
## Term: cell migration
## Ontology: BP
## Definition: The controlled self-propelled movement of a cell from one
##     site to a destination guided by molecular cues. Cell migration is
a
##     central process in the development and maintenance of multicellula
r
##     organisms.
## -------------------------------------

go.results3 <- goseq(pwf3, "hg19", "ensGene")

## Fetching GO annotations...

## For 6343 genes, we could not find any categories. These genes will be
excluded.

## To force their use, please run with use_genes_without_cat=TRUE (see do
cumentation).
```

```
## This was the default behavior for version 1.15.1 and earlier.

## Calculating the p-values...

## 'select()' returned 1:1 mapping between keys and columns

head(go.results3)

##         category over_represented_pvalue under_represented_pvalue numD
EInCat
## 938   GO:0002250           1.601071e-36                          1
44
## 12622 GO:0046649           2.326891e-34                          1
50
## 1024  GO:0002376           1.801146e-32                          1
90
## 3517  GO:0006955           2.100249e-29                          1
74
## 1194  GO:0002682           1.043842e-27                          1
63
## 12003 GO:0045321           8.395021e-26                          1
56
##       numInCat                                  term ontology
## 938        370            adaptive immune response       BP
## 12622      604               lymphocyte activation       BP
## 1024      2582               immune system process       BP
## 3517      1762                     immune response       BP
## 1194      1366 regulation of immune system process       BP
## 12003     1124                leukocyte activation       BP

enriched.GO3=go.results3$category[p.adjust(go.results3$over_represented_p
value,
                                   method="BH")<.05]

head(enriched.GO3)

## [1] "GO:0002250" "GO:0046649" "GO:0002376" "GO:0006955" "GO:0002682"
## [6] "GO:0045321"
```

Categorías GO relacionadas:

```
for(go in enriched.GO3[1:5]){
  print(GOTERM[[go]])
  cat("--------------------------------------\n")
  }

## GOID: GO:0002250
## Term: adaptive immune response
## Ontology: BP
## Definition: An immune response mediated by cells expressing specific
##     receptors for antigen produced through a somatic diversification
##     process, and allowing for an enhanced secondary response to
```

```
##     subsequent exposures to the same antigen (immunological memory).
## Synonym: acquired immune response
## Synonym: immune memory response
## -------------------------------------
## GOID: GO:0046649
## Term: lymphocyte activation
## Ontology: BP
## Definition: A change in morphology and behavior of a lymphocyte
##     resulting from exposure to a specific antigen, mitogen, cytokine,
##     chemokine, cellular ligand, or soluble factor.
## -------------------------------------
## GOID: GO:0002376
## Term: immune system process
## Ontology: BP
## Definition: Any process involved in the development or functioning of
##     the immune system, an organismal system for calibrated responses t
o
##     potential internal or invasive threats.
## -------------------------------------
## GOID: GO:0006955
## Term: immune response
## Ontology: BP
## Definition: Any immune system process that functions in the calibrated
##     response of an organism to a potential internal or invasive threat
.
## -------------------------------------
## GOID: GO:0002682
## Term: regulation of immune system process
## Ontology: BP
## Definition: Any process that modulates the frequency, rate, or extent
##     of an immune system process.
## -------------------------------------
```