PEC2_Anàlisi RNAseq amb R

Amelia Martínez Sequera

PEC2_Anàlisi RNAseq amb R

Amelia Martínez Sequera

Juny 2020 https://github.com/gititub/ADO_PEC2.RNAseqAnalysis

Continguts:

1. Introducció.

En aquest anàlisi estudiarem els perfils d'expressió a partir d´unes dades preprocessades, és a dir, l´extracció de RNA, preparació de llibreries i la seqüenciació són qüestions que no tractarem. Tanmateix, aquestes seqüències ja han estat alineades amb un genoma de referència, i s´han comptabilitzat el nombre de lectures mapades amb cada gen. Això es tradueix en una taula de recomptes (*counts.csv*), i ens interessa estudiar si aquests difereixen significativament entre els diferents grups o condicions biològiques. Ens interessa, fonamentalment, l´expressió diferencial entre 3 grups, amb 10 mostres o rèpliques cadascun, obtingudes de 54 teixits de tiroides:

- sense infiltració linfoide: NIT,
- localment infiltrat: SFI,
- extensament infiltrats: ELI.

Disposem doncs, de rèpliques tècniques i biològiques.

Les dades en brut no estan prèviament normalitzades. Això és important per seqüenciar la profunditat/mida de la biblioteca, ja que els models estadístics són més potents quan s'apliquen a recomptes no normalitzats, i estan dissenyats per tenir en compte les diferències de mida de la biblioteca (Love et Anders, 2019. *Flux de treball RNA-seq: anàlisi exploratòria a nivell de gens i expressió diferencial.*)

2. Selecció de les mostres.

A partir de l´arxiu *targets.csv* hem seleccionat 10 mostres de cada grup segons el tipus d´infiltració (NIT, SFI i ELI), sense tenir en compte el sexe ni el tipus de dades moleculars.

Tot i que ens basarem principalment en la comparació entre grups d´infiltració, hem procurat tenir mostres tant de *RNA Seq(NGS)(Next Generation Sequencing)* com d´*Allele-specific expression* (tipus de dades moleculars segons la llista de dbGap (http://www.ncbi.nlm.nih.gov/gap), base de dades de genotips i fenotips). El segon tipus fa referència a un subconjunt de gens (també SNPs i INDELs) que mostra una desviació a la presentació igual prevista dels al.lels parentals (autosòmics) i expressa preferentment l´al.lel d´un sol progenitor. En canvi, les dades tipus *RNA seq(NGS)* poden ser transcriptomes sencers, exomes,... poden incloure elements tant codificants com no codificants.

Verifiquem que les files de *Sample_Name* es corresponen amb les columnes de l´arxiu *counts.csv* i extraiem la nostra matriu de recompte, sense incorporar la primera columna perquè conté els id de la llibreria (56202 files). Nota: anomenarem "gens" a cada un dels intèrvals genòmics o elements de la llibreria (etiquetes), tot i que a priori no sabem si es tracta de gens o altres tipus de transcrits (ho analitzarem més endavant).

Per fer la selecció, simplement hem agafat els primers valors (de dalt a baix, primer les 3 columnes de l´esquerra i després les últimes columnes)de la matriu lògica. Per exemple, del grup ELI les files 29,100,3,186,211,253,167,251,146,147 de *targets.csv*, que es corresponen a aquestes columnes+1 de l´arxiu *counts.csv*.

Hem fet una nova matriu que conté només els recomptes de 30 mostres, però podem emmagatzemar els identificadors de gens (la primera columna d´EntrezGeneID). Més endavant afegirem més informació sobre anotacions sobre cada gen.

3. Anàlisi i visualització exploratòries.

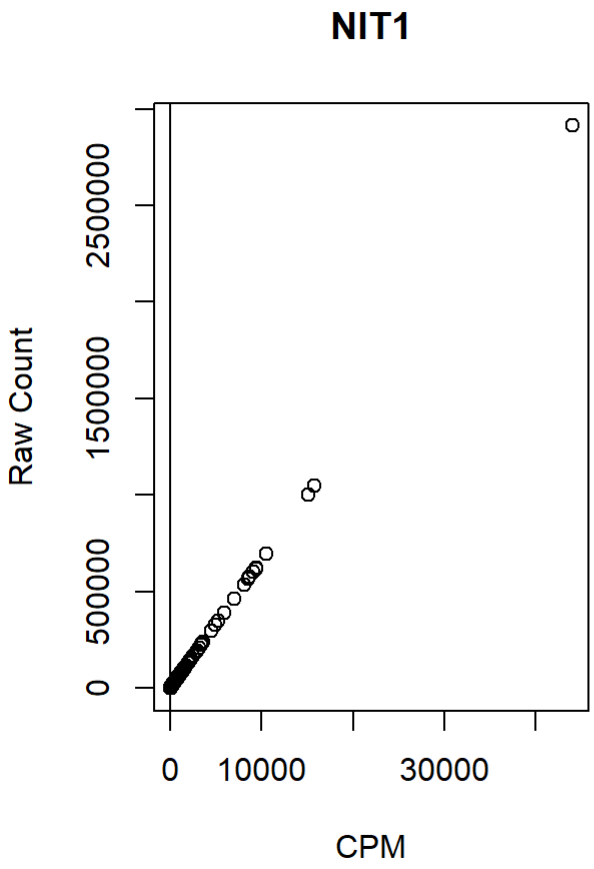
3.1. Filtrar gens amb baixa expressió.

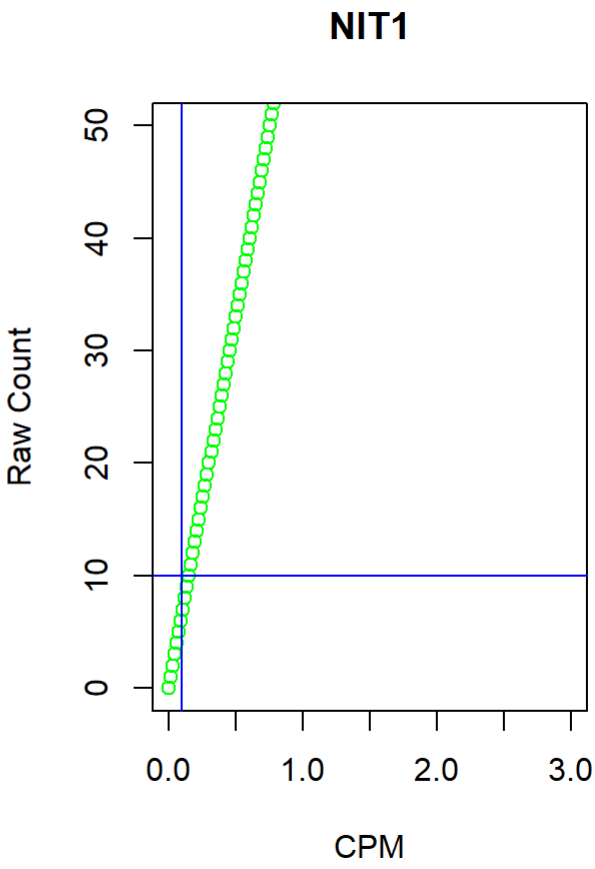
Els gens amb un recompte molt baix proporcionen poca evidència per a l'expressió diferencial i interfereixen amb algunes de les aproximacions estadístiques que s'utilitzaran posteriorment en el pipeline. També s'afegeixen a la càrrega de proves múltiples a l'hora d'estimar taxes de descobriment falses (FDR), reduint el poder per detectar gens expressats de manera diferent. Aquests gens s'han de filtrar abans d'analitzar-ne més.

En aquest conjunt de dades, optem per retenir gens si s'expressen en un recompte per milió (CPM) per sobre de 0.1 en almenys 2 mostres, ja que tenim rèpliques biològiques i tècniques de cada grup. S'utilitza un CPM de 0.1 ja que correspon a un recompte de aprox. 10 per a les mides de la biblioteca d'aquest conjunt de dades. Si el recompte és menor, es considera que és molt baix, cosa que indica que el gen associat no s'expressa en aquesta mostra. S'utilitza un requisit d'expressió en dues o més biblioteques. Filtrem amb CPMs en lloc de filtrar els recomptes directament, ja que aquest no té en compte les diferències en les mides de la biblioteca entre les mostres. Per generar els valors de CPM utilitzarem la funció cpm() de la biblioteca edgeR. (Robinson, McCarthy i Smyth, 2010) S´ha de tenir en compte que mitjançant la conversió a CPM estem normalitzant les diferents profunditats de seqüenciació de cada mostra.

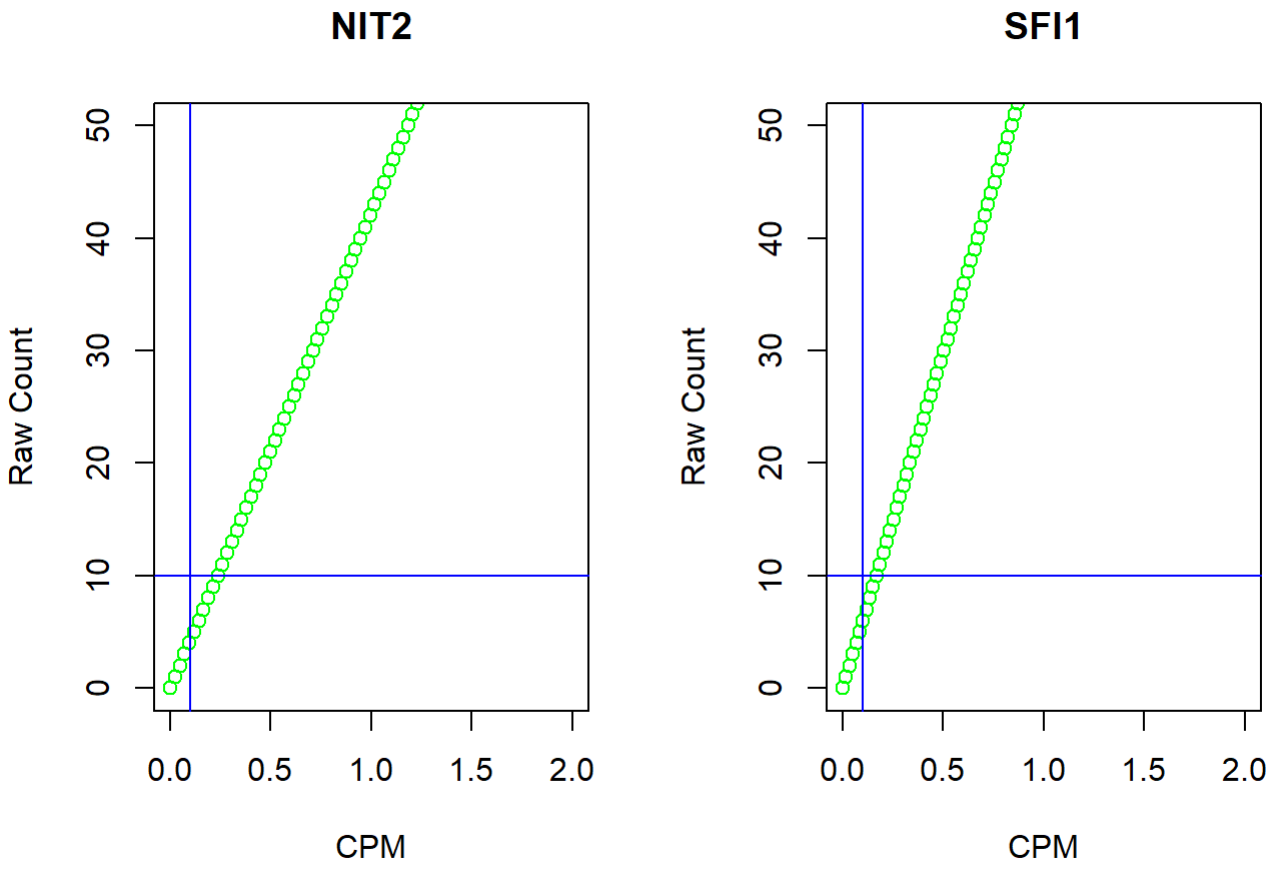
```
col1sum <- sum(mydata[,1])/1000000

mydata[1,1]/col1sum

## NIT1

## 1 0.1058108

myCPM<- cpm(mydata)

thresh<- myCPM>0.1

# Cuants TRUEs hi ha?

table(rowSums(thresh))

##
```

```
## 0 1 2 3 4 5 6 7 8 9 10 11 12
## 24203 2406 1272 863 649 507 499 447 374 380 311 314 276
## 13 14 15 16 17 18 19 20 21 22 23 24 25
## 269 276 261 278 271 269 259 249 259 316 304 306 316
## 26 27 28 29 30
## 360 435 563 856 17854
# mantenim el gens que tenen, al menys, 2 TRUES en cada fila de thresh
keep <- rowSums(thresh) >= 2
# ens quedem amb 29593
counts.keep <- mydata[keep,]
summary(keep)
## Mode FALSE TRUE
## logical 26609 29593
dim(counts.keep)
## [1] 29593 30
# Comprovem si el nostre llindar de 0.1 correspon a un recompte aproximat a
10. Mirem la primera mostra:
par(mfrow=c(1,2))
plot(myCPM[,1], mydata$NIT1, xlab="CPM", ylab="Raw Count",
main=colnames(myCPM)[1])
abline(v=0.1)
plot(myCPM[,1],mydata$NIT1, xlab="CPM", ylab="Raw Count", main=colnames(myCPM)[1],
ylim=c(0,50), xlim=c(0,3), col="green")
abline(v=0.1, h=10,col="blue")
```






```
par(mfrow=c(1,2))
plot(myCPM[,2],mydata$NIT2, xlab="CPM", ylab="Raw Count", main=colnames(myCPM)[2],
ylim=c(0,50), xlim=c(0,2), col="green")
abline(v=0.1, h=10,col="blue")
plot(myCPM[,11],mydata$SFI1, xlab="CPM", ylab="Raw Count", main=colnames(myCPM)[11],
ylim=c(0,50), xlim=c(0,2), col="green")
abline(v=0.1, h=10,col="blue")
```



Els gens amb un valor CPM inferior a un determinat cutoff es considera que és molt baix, cosa que indica que el gen associat no s'expressa en aquesta mostra. Els llindars de CPM més petits solen ser adequats per a biblioteques més grans.

3.2. Conversió a DGEList.

A continuació, crearem un objecte *DGEList*. Aquest és un objecte utilitzat per *edgeR* per emmagatzemar dades de recompte. Disposa de diversos "slots" (ranures) per emmagatzemar diversos paràmetres sobre les dades.La informació sobre la mida de la biblioteca es desa a la ranura mostral.

```
dgeObj <- DGEList(counts.keep)
```

```
dgeObj
```

```
## An object of class "DGEList"
```

$counts

NIT1 NIT2 NIT3 NIT4 NIT5 NIT6 NIT7 NIT8 NIT9 NIT10 SFI1 SFI2 SFI3 SFI4 SFI5

1 7 1 0 1 3 2 0 1 2 17 1 0 3 2 9

2 401 768 719 481 781 1517 629 720 542 484 1164 633 426 679 302

3 4 0 1 0 0 0 1 0 1 10 2 2 1 2 4

4 2 1 1 1 2 0 1 1 0 7 2 1 1 4 2

5 6 10 3 10 1 5 8 4 6 4 19 14 5 0 10

SFI6 SFI7 SFI8 SFI9 SFI10 ELI1 ELI2 ELI3 ELI4 ELI5 ELI6 ELI7 ELI8 ELI9 ELI10

1 3 2 1 3 2 3 0 1 3 3 4 5 1 1 0

2 1704 491 874 533 457 1301 1002 474 134 979 1325 489 1472 775 834

3 1 2 8 1 1 1 1 1 1 3 1 1 1 2 1

4 1 0 2 0 2 0 0 0 2 2 0 3 0 0 1

5 5 7 4 0 9 5 15 3 3 4 4 7 38 10 6

29588 more rows ...

##

$samples

group lib.size norm.factors

NIT1 1 66150219 1

NIT2 1 42266701 1

NIT3 1 50694521 1

NIT4 1 47489104 1

NIT5 1 60977255 1

25 more rows ...

3.3. Control de qualitat

Anema a comprovar que les dades són de bona qualitat i que les mostres són com esperaríem.

Mida de la llibreria.

Primer, podem comprovar quantes lectures tenim per a cada mostra del dgeObj.

dgeObj$samples

group lib.size norm.factors

NIT1 1 66150219 1

NIT2 1 42266701 1

NIT3 1 50694521 1

NIT4 1 47489104 1

NIT5 1 60977255 1

NIT6 1 46699715 1

NIT7 1 41662275 1

NIT8 1 55893949 1

NIT9 1 39639702 1

NIT10 1 69802989 1

SFI1 1 59529266 1

SFI2 1 68482369 1

SFI3 1 51872996 1

SFI4 1 40855745 1

SFI5 1 39858001 1

SFI6 1 84713691 1

SFI7 1 45883365 1

SFI8 1 53124121 1

SFI9 1 44340733 1

SFI10 1 55878027 1

ELI1 1 51970092 1

ELI2 1 61440249 1

ELI3 1 52193544 1

ELI4 1 15481426 1

ELI5 1 85620372 1

ELI6 1 73979605 1

ELI7 1 64433651 1

ELI8 1 81214046 1

ELI9 1 64698354 1

ELI10 1 42006161 1

#dgeObj$samples$lib.size

#dgeObj$samples[,2]

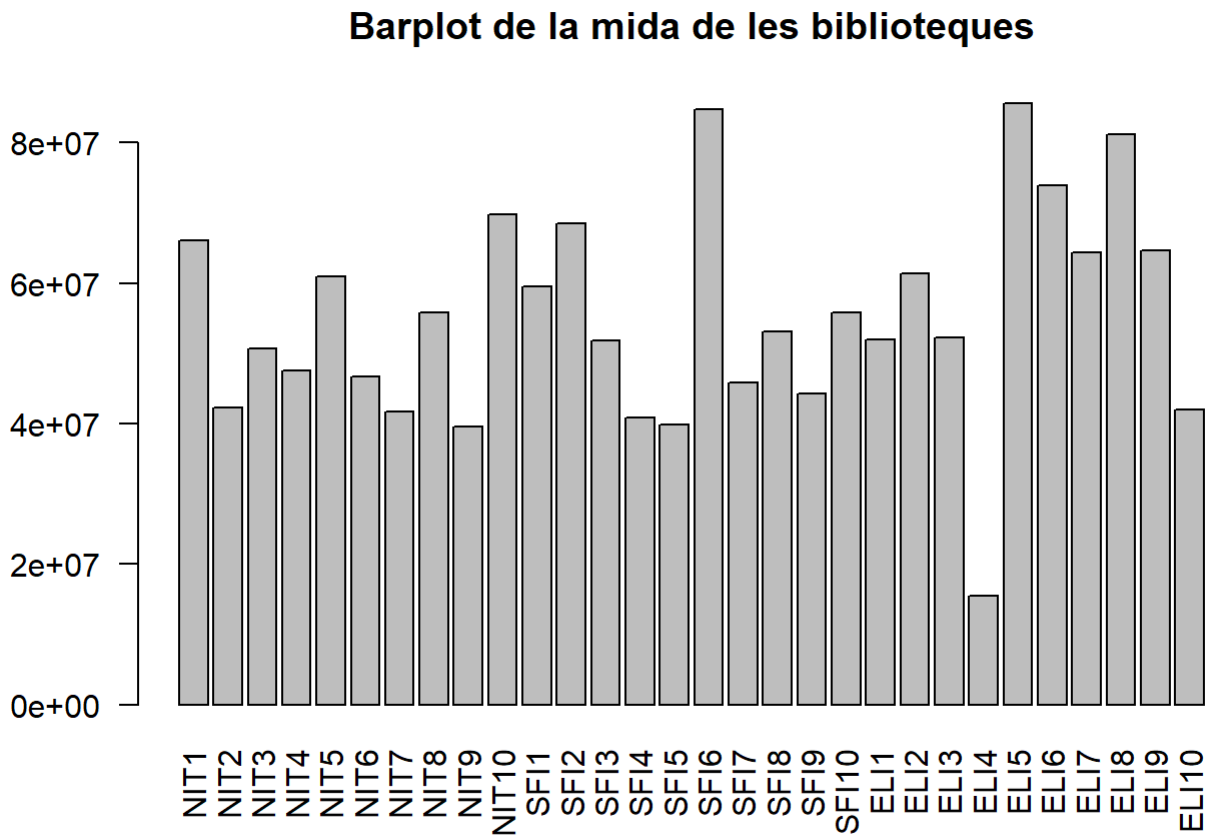
#dgeObj$samples[,"lib.size"]

Gràfica de densitat.

Podem fer una gràfica de barres per veure amb més facilitat si hi ha discrepàncies importants entre les mostres.

barplot(dgeObj$samples$lib.size, names=colnames(dgeObj), las=2)

title("Barplot de la mida de les biblioteques")



Boxplots.

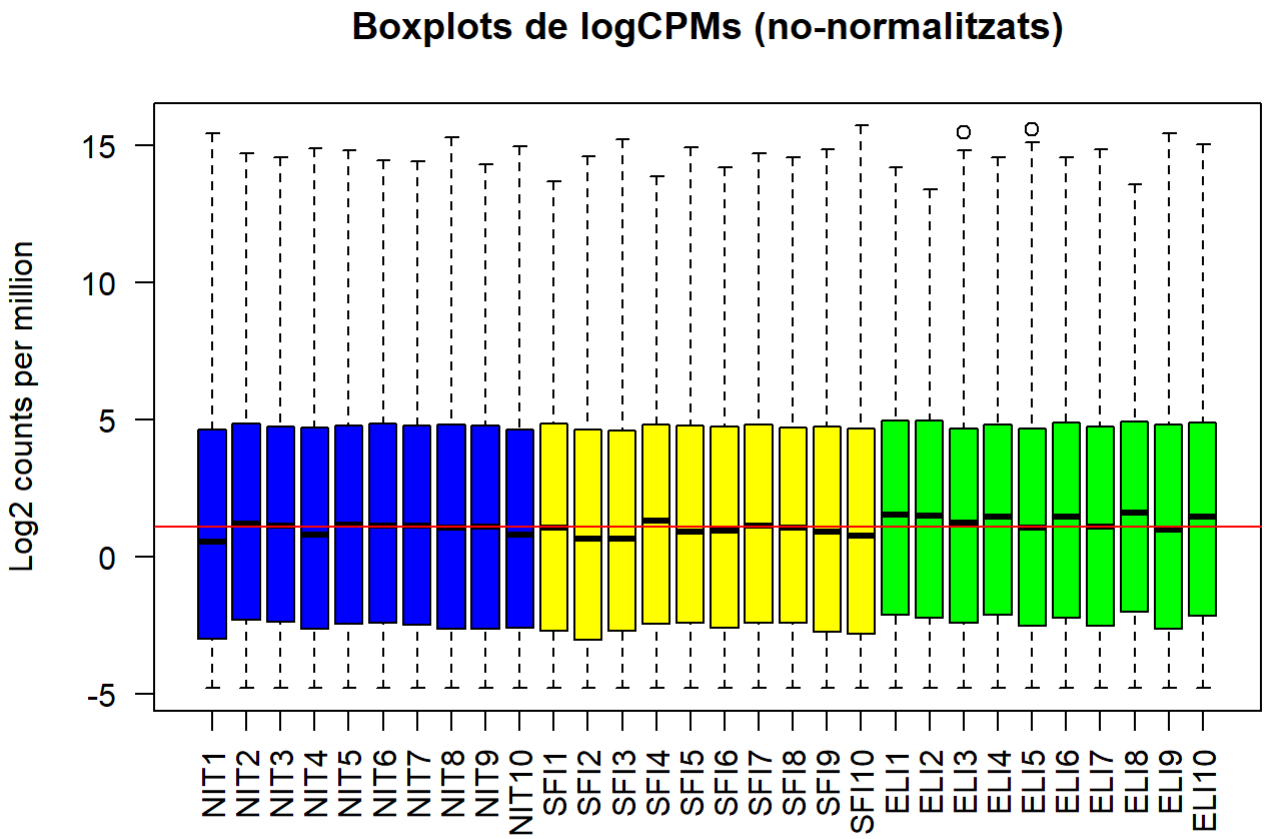
Les dades de recompte no es distribueixen normalment. Utilitzarem boxplots per comprovar la distribució dels recomptes a l'escala log2. Podem utilitzar la funció cpm per obtenir recomptes de log2 per milió, que es corregeixen per a les diferents mides de la biblioteca. La funció de cpm també afegeix una petita compensació per evitar tenir registre de zero.

logcounts <- cpm(dgeObj,log=TRUE)

boxplot(logcounts, xlab="", ylab="Log2 counts per million",las=2, col=c(rep("blue",10), rep("yellow",10),rep("green",10)))

abline(h=median(logcounts),col="red")

title("Boxplots de logCPMs (no-normalitzats)")

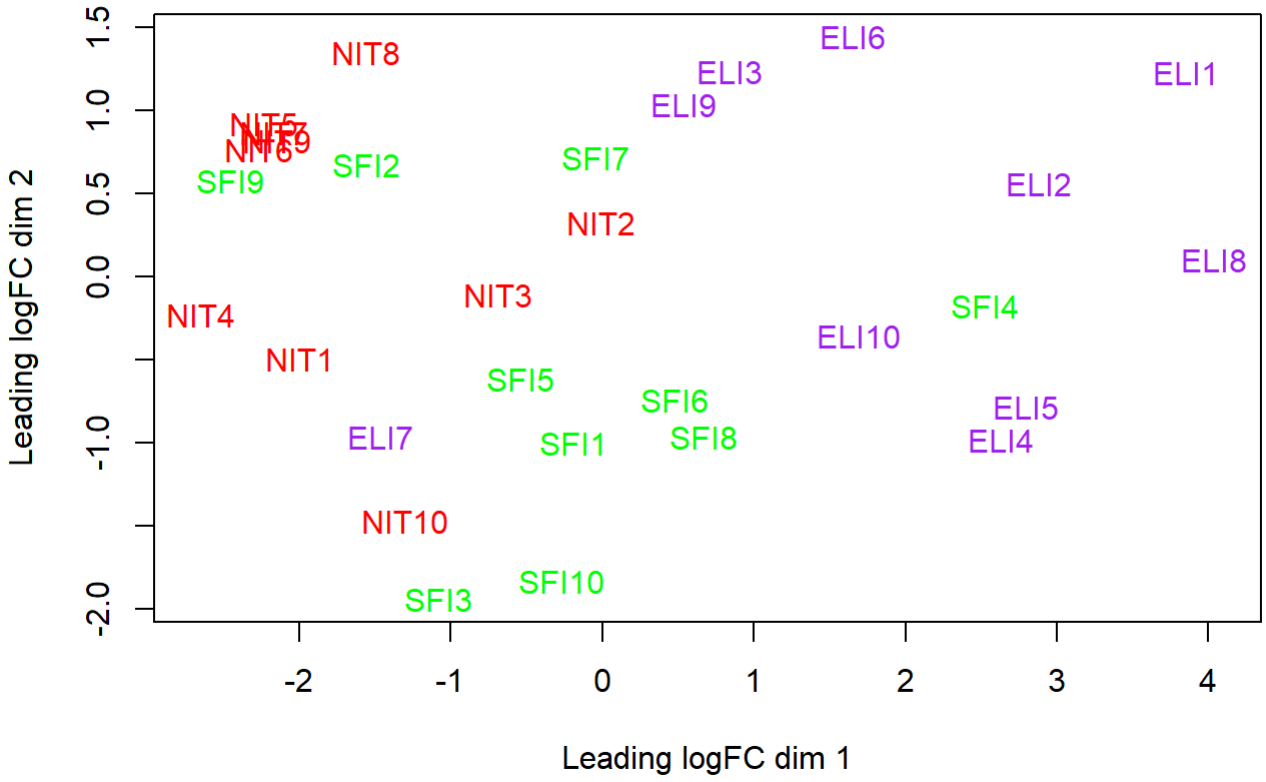


Observem que les distribucions no són idèntiques però la mitjana s´ajusta bastant a la linia vermella, que correspòn a la mitjana de logCPM.

Gràfiques d'escalació multidimensional (MDS).

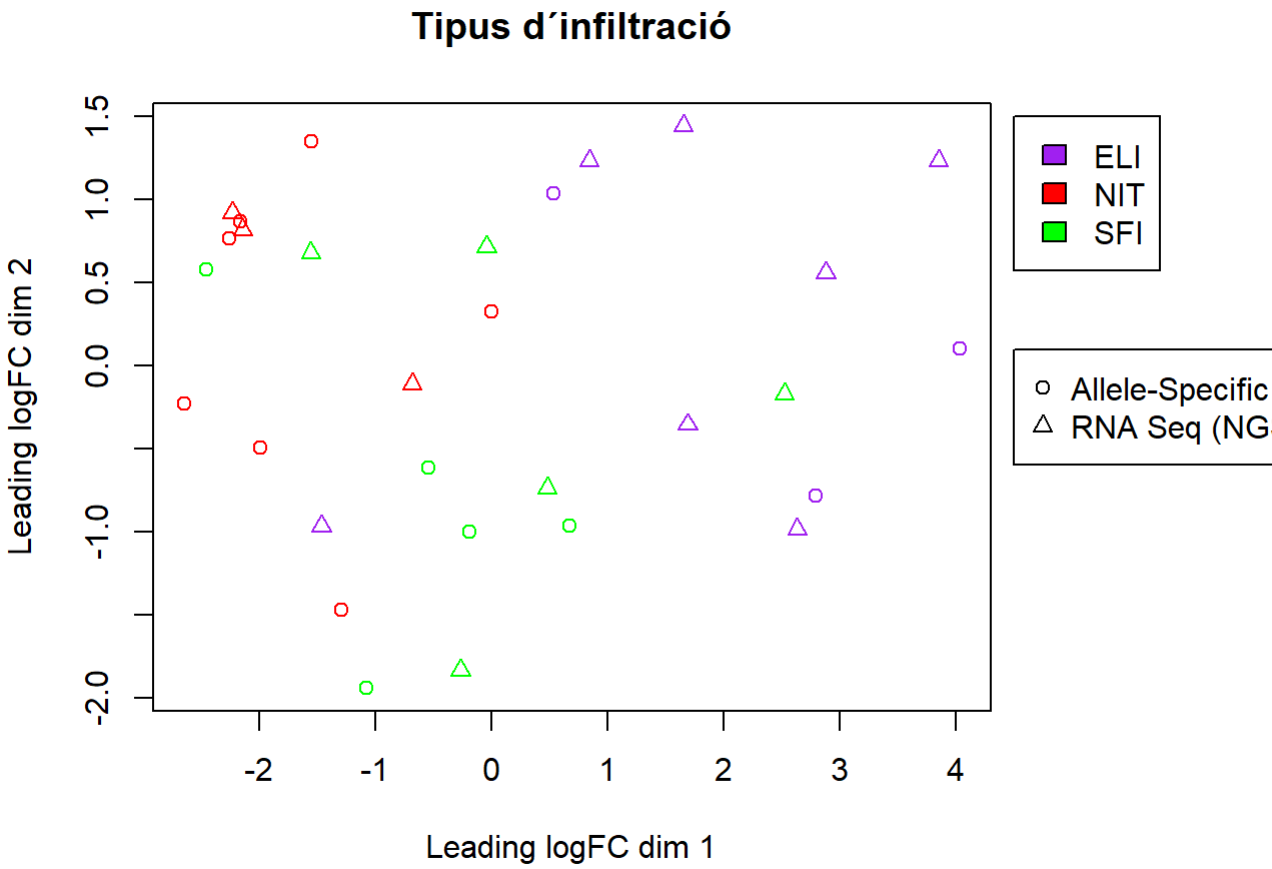
Una MDSplot és una visualització d'un PCA (anàlisi de components principals), que determina les majors fonts de variació de les dades. Un PCA és un exemple d'anàlisi no supervisada, en què no cal especificar els grups. Si el nostre experiment està ben controlat i ha funcionat bé, el que esperem veure és que les fonts més grans de variació de les dades són els tractaments/grups que ens interessen. També és una eina molt útil per controlar la qualitat i comprovar els outliers.

plotMDS(dgeObj, col=col.cell)



Les mostres del mateix grup s´haurien de veure més juntes. Veiem que les mostres del grup ELI s´agrupen al cuadrant superior dret, excepte ELI7. NIT i SFI clarament es posicionen a l´esquerra, excepte SFI4. Si a més tenim en comte el tipus de dada:

```
par(xpd= T, mar = par()$mar + c(0,0,0,7))
plotMDS(dgeObj,col=col.cell,pch =c(1,2)[mol.type])
legend( 4.5,1.5,fill=c("purple","red","green"),legend=levels(group))
legend(4.5,0.1, legend= levels(mol.type),pch=c(1,2))
title("Tipus d´infiltració")
```



par(mar=c(5, 4, 4, 2) + 0.1)

El que s´espera veure és que les mostres d'un mateix grup de grups apareguin més juntes, mentre que mostres de diferents grups formen agrupaments separats. Això indica que les diferències entre grups són més grans que dintre dels grups, és a dir, que l'expressió diferencial és més gran que la variància i es pot detectar. A la gràfica podem veure que la distància entre mostres NIT de l´esquerra i les mostres ELI de la dreta és d´aprox. una unitat, el que correspon a un canvi(fold change) de $2 \hat{} 1 = 2$ entre els dos grups.

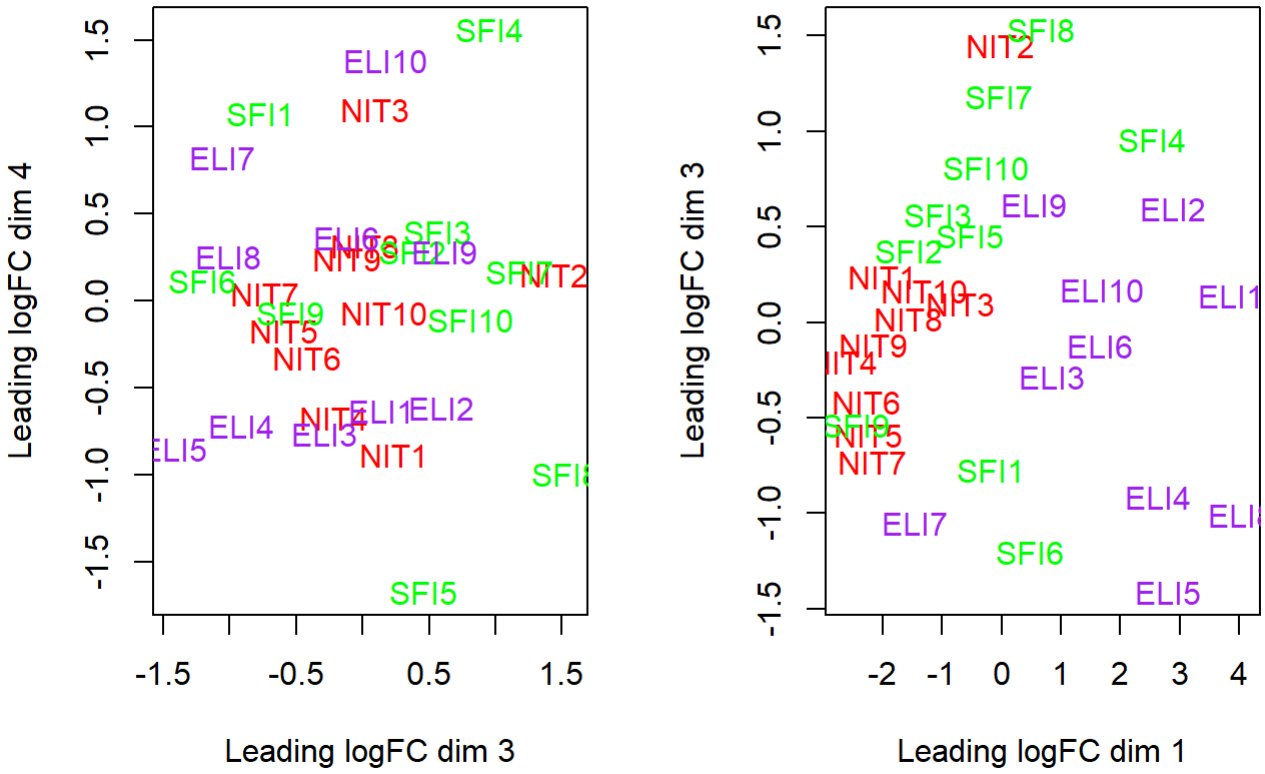
L'agrupament a la gràfica MDS es pot utilitzar per motivar canvis a la matriu de disseny a la llum dels possibles efectes del lot. Per exemple, si la primera rèplica de cada grup es va preparar en un moment diferent de la segona rèplica. Si la gràfica MDS mostrava la separació de les mostres per temps, pot valdre la pena incloure temps a l'anàlisi del flux descendent per tenir en compte l'efecte basat en el temps.

plotMDS traça de manera predeterminada les dues primeres dimensions, però podem graficar d´altres:

par(mfrow=c(1,2))

plotMDS(dgeObj,dim=c(3,4), col=col.cell)

plotMDS(dgeObj,dim=c(1,3), col=col.cell)



Una altra alternativa és fer una gràfica MDS interactiva mitjançant el paquet Glimma. Això ens permet explorar de forma interactiva les diferents dimensions. Glimma es va crear per fer versions interactives d'alguns dels gràfics més populars del paquet limma. Actualment, es pot utilitzar per obtenir gràfiques MDS i gràfics MD. La sortida de glMDSPlot és una pàgina html (/mds/MDS-Plot.html) que mostra la trama MDS a l'esquerra i la quantitat de variació explicada per cada dimensió en una barra de bar a la dreta. L'usuari pot passar per punts per obtenir informació de mostra i canviar entre les dimensions successives de la

trama MDS fent clic a les barres de la barra de comandament. Les gràfiques MDS predeterminades mostren les dimensions 1 i 2.

labels <- paste(targets2$Sample_Name, group, mol.type)

group <- paste(group,mol.type,sep=".")

glMDSPlot(dgeObj, labels=labels, groups=group, folder="mds")

html adjunt a la carpeta mds.

Agrupació jeràrquica (clustering) amb heatmaps.

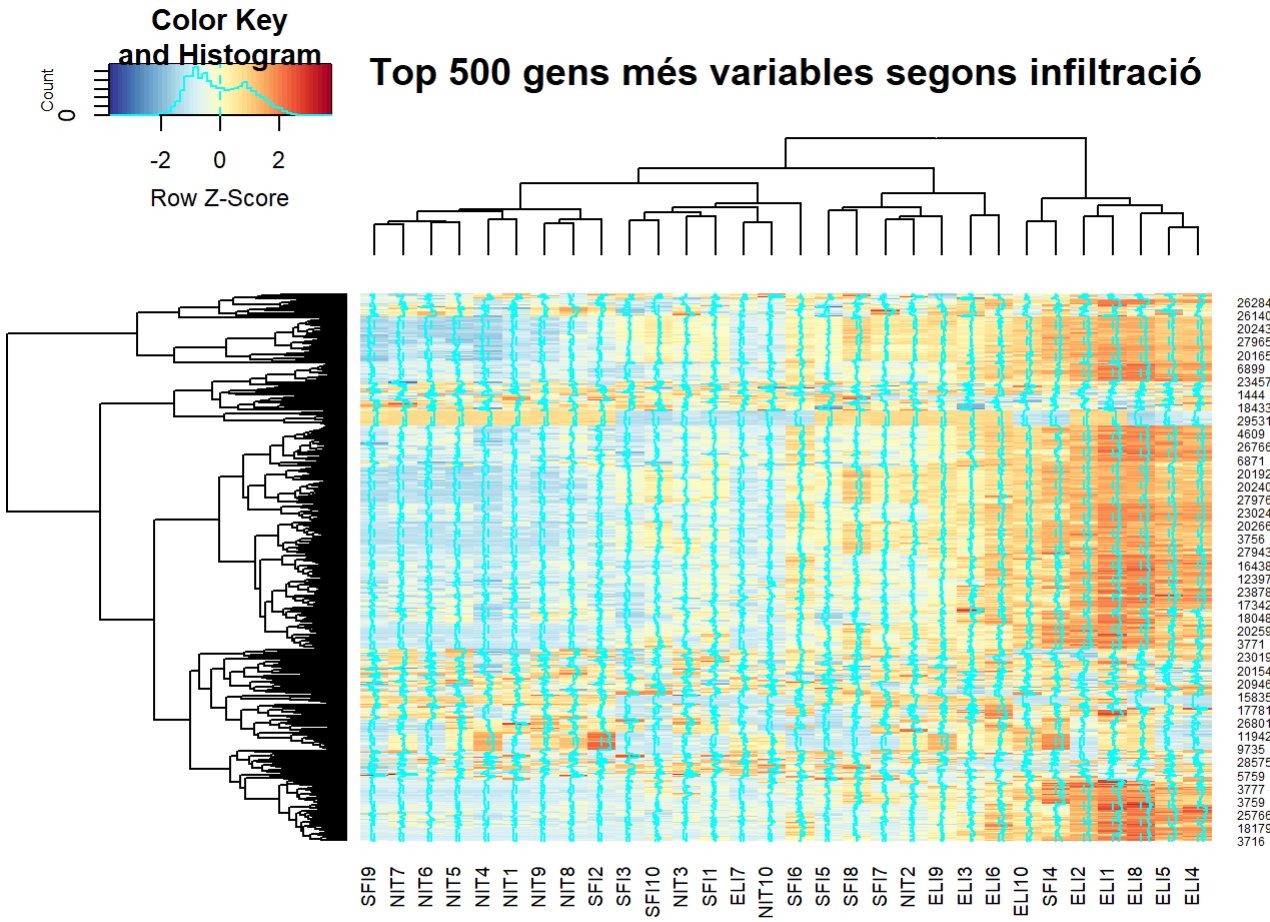
Seleccionem les dades dels 500 gens més variables i traçem el mapa de calor

1 2 3 4 5 6

0.6438267 0.3458818 0.3849616 0.3793817 0.8120733 2.3458088

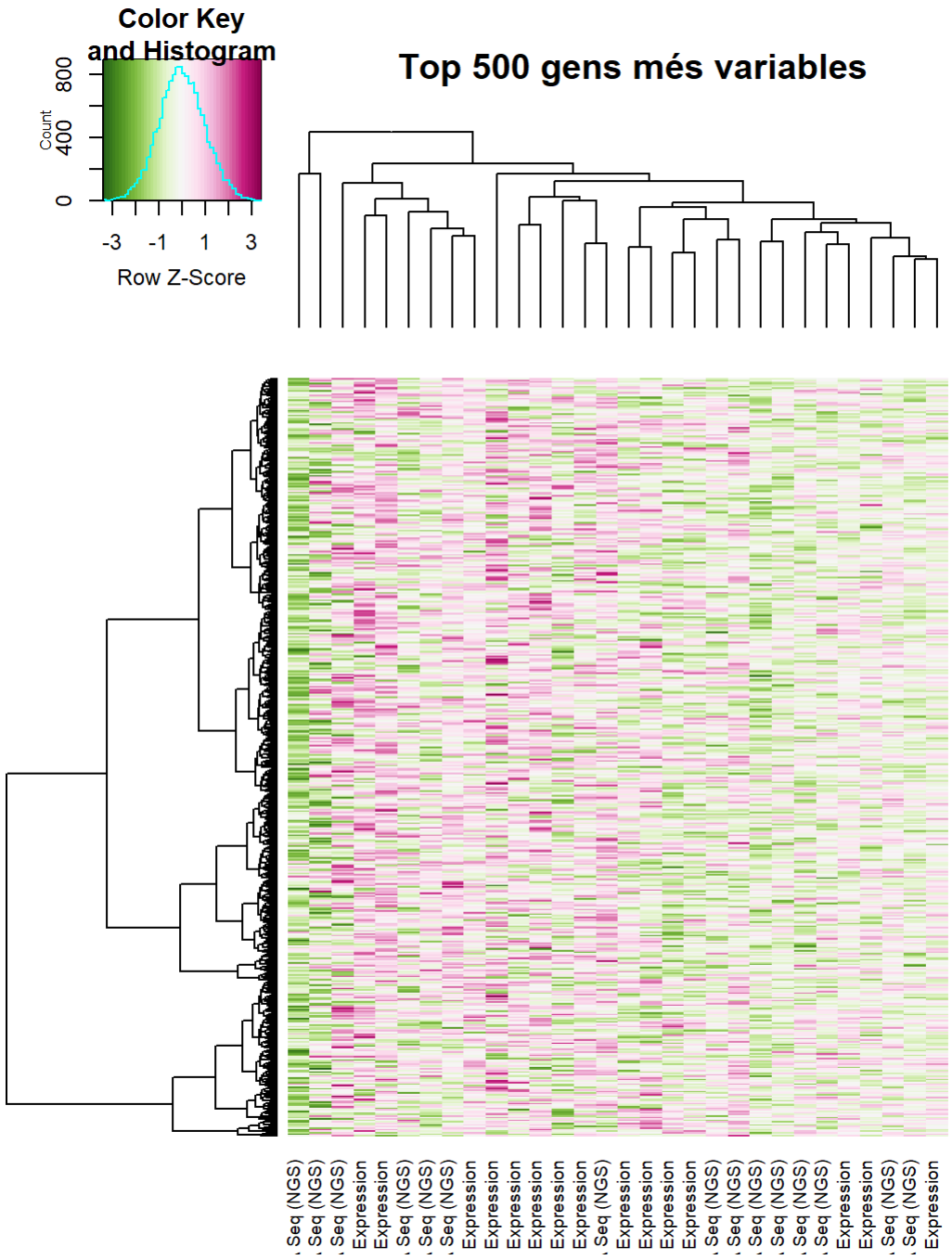
[1] "29028" "29559" "29562" "29545" "29546" "29528"

[1] 500 30


Color Key
and Histogram
Count
0
-2
0
2
Row Z-Score
Top 500 gens més variables segons infiltració
SFI9
NIT7
NIT6
NIT5
NIT4
NIT1
NIT9
NIT8
SFI2
SFI3
SFI10
NIT3
SFI1
ELI7
NIT10
SFI6
SFI5
SFI8
SFI7
NIT2
ELI9
ELI3
ELI6
ELI10
SFI4
ELI2
ELI1
ELI8
ELI5
ELI4

png

2


Color Key
and Histogram
Count
0
400
800
-3
-1
1
3
Row Z-Score
Top 500 gens més variables
Seq (NGS)
Expression

3.4. Normalització per biaix de composició.

El mètode TMM (trimmed mean of M-values normalization method) es realitza per eliminar els biaixos de composició entre biblioteques (Mark D Robinson i Oshlack 2010). Això genera un conjunt de factors de normalització, on el producte d'aquests factors i les mides de la biblioteca defineixen la mida efectiva de la biblioteca.

La funció calcNormFactors en edgeR calcula els factors de normalització entre llibreries. TMM, i la major part dels mètodes de normalització, escalen en realció d una mostra. Això actualitzarà els factors de normalització de l'objecte DGEList (els seus valors per defecte són 1). Mirem els factors de normalització d'aquestes mostres:

```
dgeObj <- calcNormFactors(dgeObj)
```

```
head(dgeObj$samples)
```

```
## group lib.size norm.factors
## NIT1 1 66150219 0.8516386
## NIT2 1 42266701 1.0908100
## NIT3 1 50694521 1.0170699
## NIT4 1 47489104 0.9616483
## NIT5 1 60977255 1.0467420
## NIT6 1 46699715 1.0779419
```

```
sort(dgeObj$samples$norm.factors, decreasing = T)
```

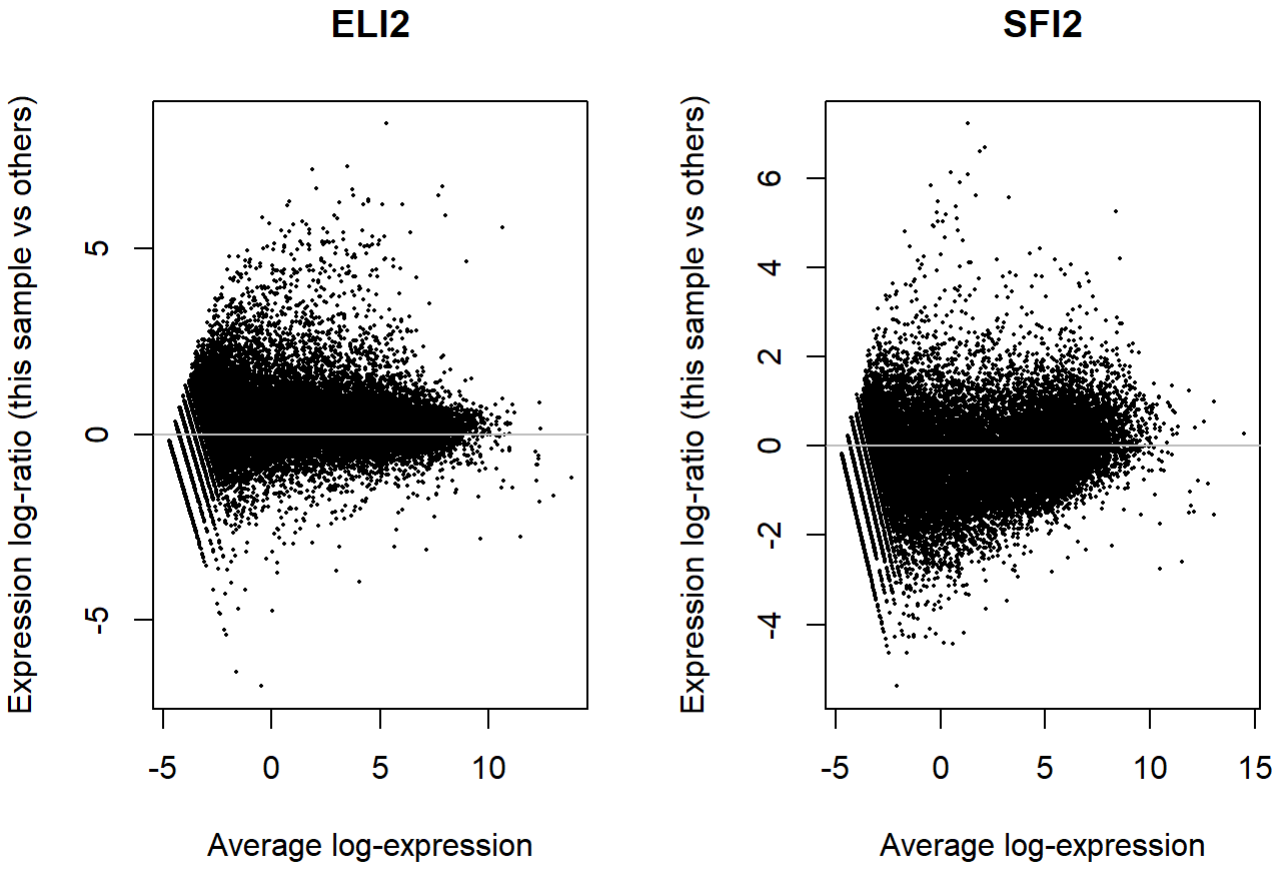
```
## [1] 1.1075602 1.0980930 1.0952598 1.0918830 1.0908100 1.0810463 1.0779419
## [8] 1.0756108 1.0467420 1.0379452 1.0346108 1.0325596 1.0245074 1.0170699
## [15] 1.0092566 1.0049938 0.9989586 0.9987070 0.9971926 0.9912672 0.9893356
## [22] 0.9788644 0.9616483 0.9388134 0.9159077 0.9079877 0.8982797 0.8913085
## [29] 0.8516386 0.8384465
```

Els factors de normalització multipliquen per unificar totes les llibreries. Un factor de normalització inferior a 1 indica que es reduirà la mida de la biblioteca, ja que hi ha més supressió (és a dir, biaix de composició) respecte a les altres biblioteques. Això equival a escalar els recomptes cap amunt en aquesta mostra. Per contra, un factor superior a 1 augmentarà de mida de la biblioteca i equivaldrà a reduir els recomptes.

SF2 i NIT1 presenten els factors de normalització menors, i els més grans són ELI2 i ELI10.Si traçem la diferència mitjana mitjançant la funció plotMD per a aquestes mostres, hauríem de ser capaços de veure el problema de biaix de la

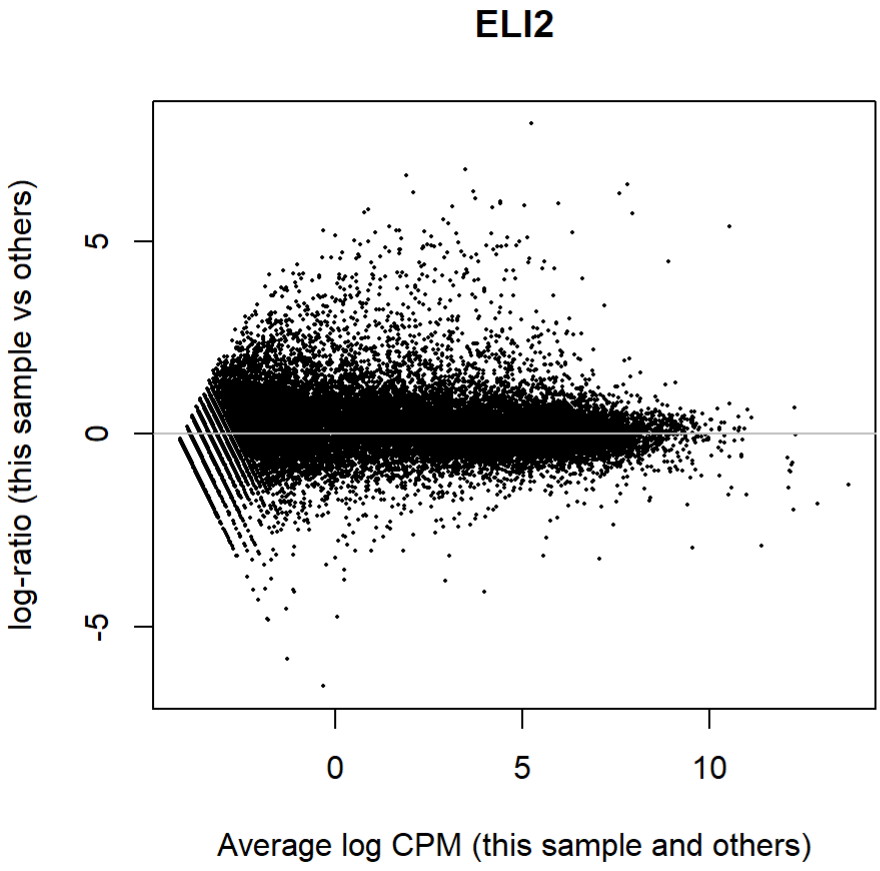
composició. Utilitzarem els "logcounts", que s'han normalitzat per la mida de la biblioteca, però no per al biaix de composició.

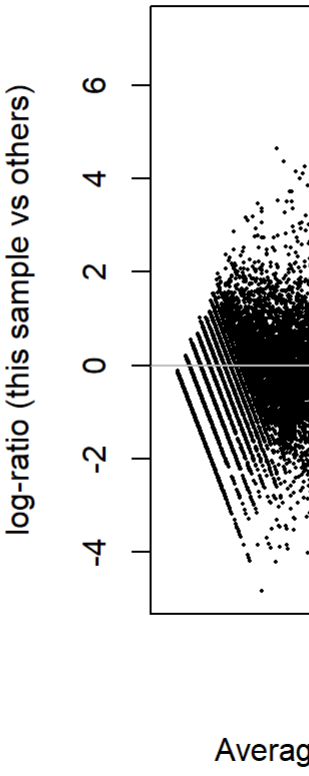
```
par(mfrow=c(1,2))
plotMD(logcounts,column = 22)
abline(h=0,col="grey")
plotMD(logcounts,column = 12)
abline(h=0,col="grey")
```



Les gràfiques MD mostren una expressió mitjana (mitjana: eix x) contra canvis *log-fold* (diferència: eix y). Com que la nostra DGEList conté els factors de normalització, si féssim aquestes gràfiques mitjançant dgeObj, hauríem de veure que s'ha resolt el problema de biaix de composició.

```
par(mfrow=c(1,2))
plotMD(dgeObj,column = 22)
abline(h=0,col="grey")
plotMD(dgeObj,column = 12)
abline(h=0,col="grey")
```



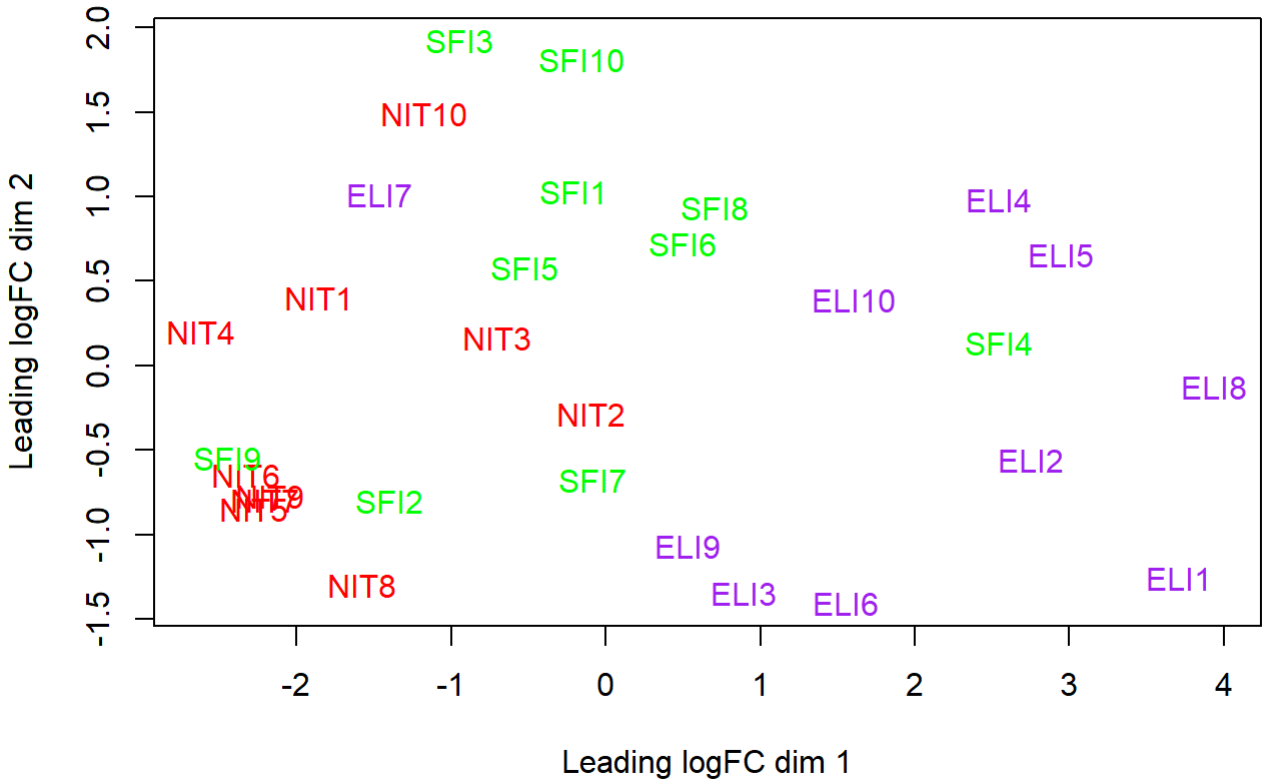


4. Anàlisi de l´expressió diferencial.

4.1. Exploració de dades.

Una gràfica MDS mostra distàncies, en termes de coeficient biològic de variació (BCV), entre mostres.Ens dóna informació sobre la qualitat de les dades.

plotMDS(dgeObj,col=col.cell)



4.2. Estimació de la dispersió.

El primer pas important en l'anàlisi de dades de DGE és estimar el paràmetre de dispersió de cada gen, una mesura del grau de variació "interbibliotecària" per a aquest gen. L'estimació de la dispersió comuna dóna una idea de la variabilitat global a través del genoma d'aquest conjunt de dades, promediada per a tots els gens:

d1 <- estimateCommonDisp(dgeObj, verbose = T)

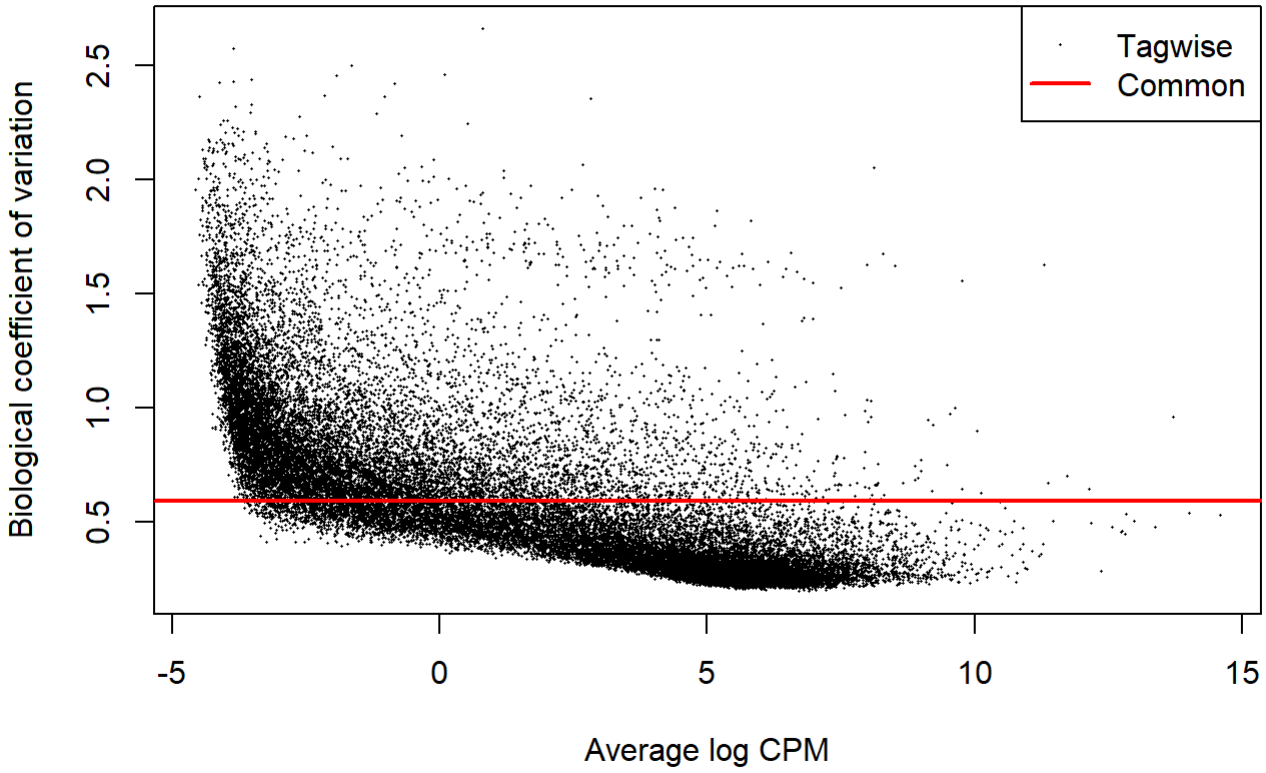
Disp = 0.34592 , BCV = 0.5882

Per a l'anàlisi d'expressions diferencials rutinàries, s'utilitzen dispersions empíriques bayesianes amb etiquetes. Cal estimar la dispersió comuna abans

d'estimar les dispersions amb les etiquetes.

d1 <- estimateTagwiseDisp(d1)

plotBCV(d1)



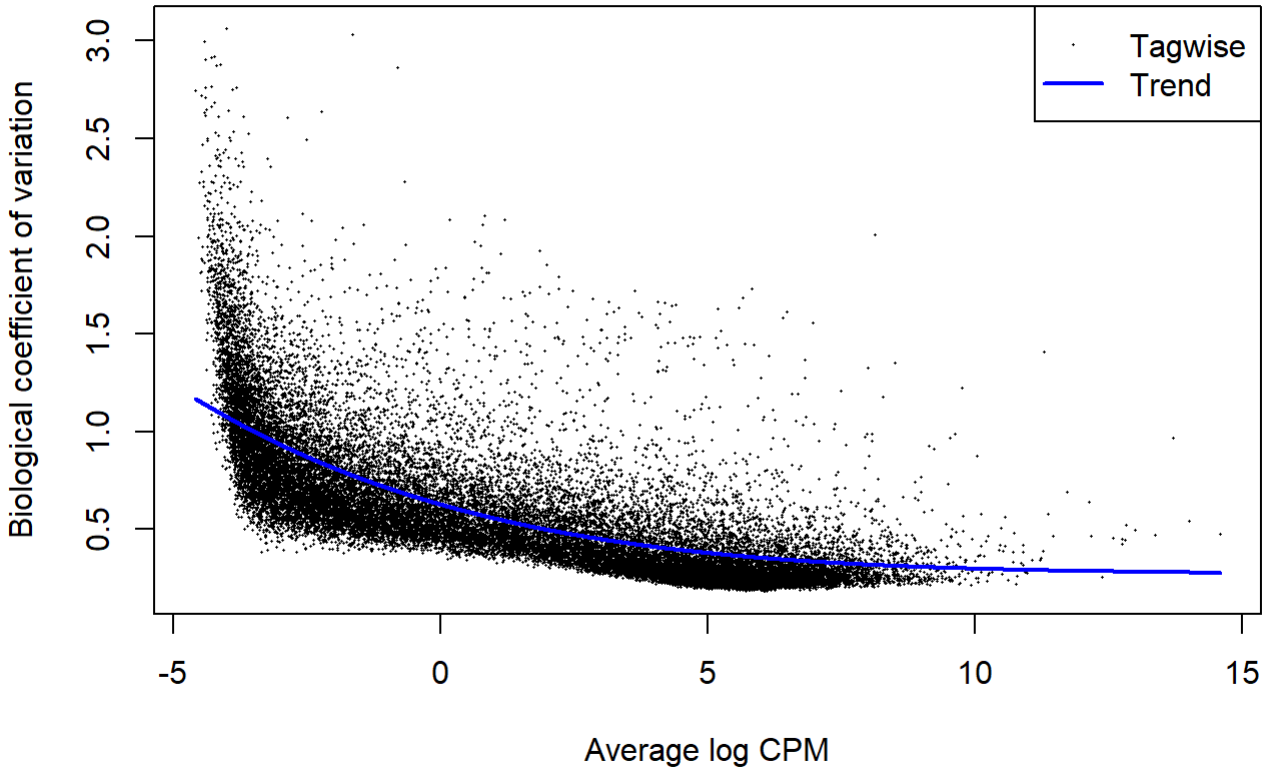
Estimacions de dispersió GLM.

Per encaixar un model a edgeR hem de seguir diversos passos. En primer lloc, cal adaptar-se a la dispersió comuna. Després cal adaptar-se a un model de tendència (si no s'adapta a una tendència, el valor per defecte és utilitzar la dispersió comuna com a tendència). A continuació, podem encaixar la dispersió en etiquetes, que és una funció d'aquest model.

A més del dispersió comú i sense etiquetes, també podem estimar un model lineal generalitzat (glm) adequat amb edgeR. A continuació, farem estimacions de

dispersió genètiques, permetent una possible tendència amb la mida del recompte de mitjanes. plotBCV() traça el coeficient biològic de variació (arrel quadrada de les dispersions) contra log2-CPM:

```
dgeObj$samples$group=c(rep("1",10),rep("2",10),rep("3",10))

design.mat <- model.matrix(~ 0 + dgeObj$samples$group)

colnames(design.mat) <- levels(dgeObj$samples$group)

d2 <- estimateGLMCommonDisp(dgeObj,design.mat)

d2 <- estimateGLMTrendedDisp(dgeObj,design.mat, method="power")

# Es pot canviar el "method" a "auto", "bin.spline", "power", "spline", "bin.loess".

# El mètode per defecte es "auto" que fa servir "bin.spline" per >200 etiquetes, i "power" en els altres casos.

d2 <- estimateGLMTagwiseDisp(d2,design.mat)

plotBCV(d2)
```



Comparació dels models en DESeq i edgeR.

Segons quin model fem servir s´obtenen diferents resultats. A la gràfica tenim dispersió a l´eix vertical, en comptes del BCV.

```
cds <- newCountDataSet(data.frame(dgeObj$counts), dgeObj$samples$group)

cds <- estimateSizeFactors( cds )

sizeFactors( cds )

## NIT1 NIT2 NIT3 NIT4 NIT5 NIT6 NIT7 NIT8

## 1.0773554 0.8667826 0.9688980 0.8700072 1.2195433 0.9448486 0.8106776
1.0626963

## NIT9 NIT10 SFI1 SFI2 SFI3 SFI4 SFI5 SFI6
```

0.7693047 1.1915316 1.1666598 1.0926211 0.8842804 0.7695018 0.7505160
1.6004116

SFI7 SFI8 SFI9 SFI10 ELI1 ELI2 ELI3 ELI4

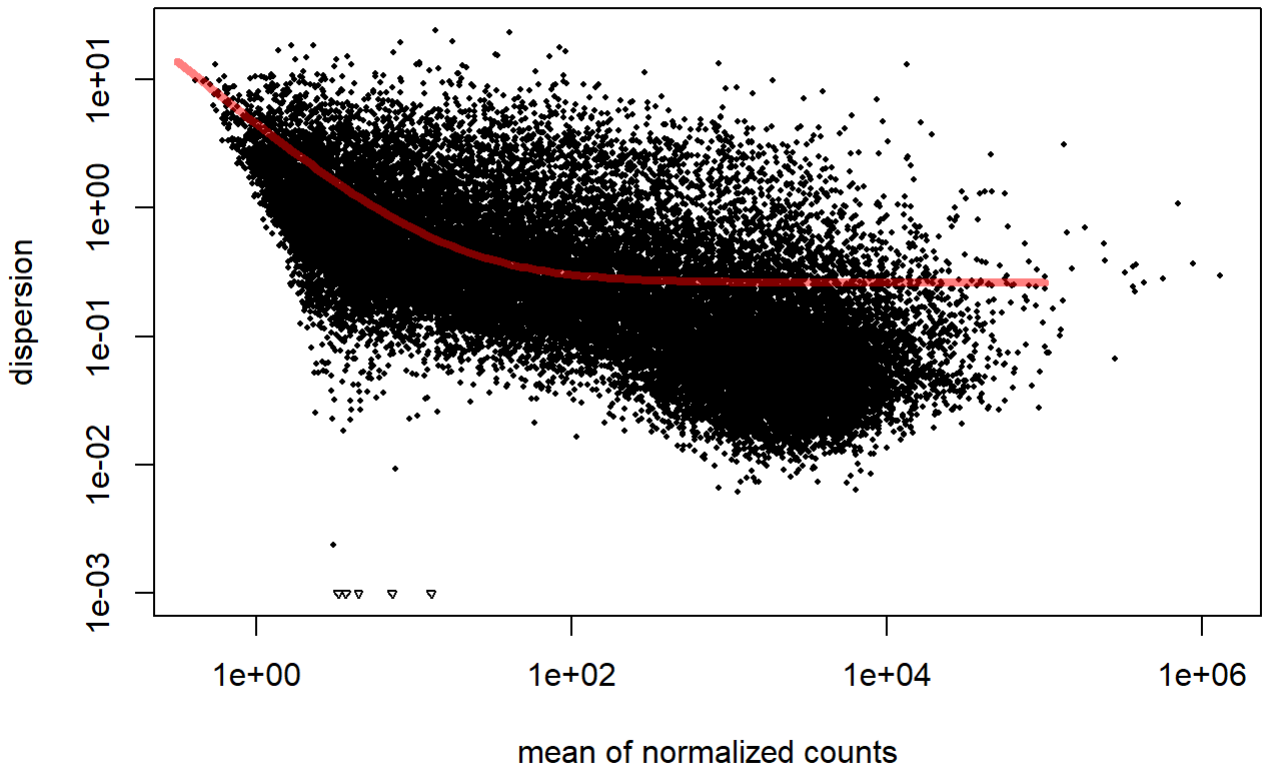
0.8980956 0.9871033 0.8325594 0.9733940 1.0667781 1.3229728 0.9316612
0.3036155

ELI5 ELI6 ELI7 ELI8 ELI9 ELI10

1.4690830 1.5358734 1.2381607 1.6835642 1.2218092 0.8802889

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

```
plotDispEsts(cds)
```



Dispersió i variació biològica.

La dispersió d'un gen és simplement una altra mesura de la variància d'un gen i és usat per DESeq per modelar la variància global dels valors de recompte d'un gen. El model per a la variància (v) dels valors de recompte utilitzat per DESeq és:

[v = s mu + alpha s ^ 2 mu ^ 2 text {On} alpha text {és la dispersió,} mu text {és el valor de recompte normalitzat previst i} s text { és el factor de mida}]

La dispersió es pot interpretar com el quadrat del coeficient de variació biològica (per exemple, la diferència de recomptes entre dues rèpliques biològiques és del 40%, de manera que la dispersió del gen és (0,4 ^ 2 = 0,16)).

4.3. Expressió diferencial.

La funció exactTest() fa servir el test binomial negatiu. La funció topTags() mostra convenientment els resultats més significatius. Per defecte, s'utilitza l'algoritme de Benjamini i Hochberg per controlar la taxa de descobriment falsa (FDR).

d1$samples$group=c(rep("1",10),rep("2",10),rep("3",10))

d2$samples$group=c(rep("1",10),rep("2",10),rep("3",10))

et12 <- exactTest(d1, pair=c(1,2)) # compara grups 1 i 2, NIT-SFI

et13 <- exactTest(d1, pair=c(1,3)) # compara grups 1 i 3, NIT-ELI

et23 <- exactTest(d1, pair=c(2,3)) # compara grups 2 i 3, SFI-ELI

topTags(et12, n=10)

Comparison of groups: 2-1

logFC logCPM PValue FDR

3771 7.950778 -0.04405858 9.884292e-06 0.2925058

27983 5.915688 4.73306210 3.633153e-05 0.4780336

20189 5.386518 4.20474549 6.593965e-05 0.4780336

1887 3.581459 0.54417147 1.132363e-04 0.4780336

20192 5.991051 4.39823420 1.174725e-04 0.4780336

20272 6.021113 -2.60027976 1.477671e-04 0.4780336

3759 5.971675 -0.53233877 1.526265e-04 0.4780336

14323 2.078259 5.18198115 1.907459e-04 0.4780336

27981 4.689037 -1.40146245 2.061982e-04 0.4780336

3754 5.474625 -0.30014233 2.130449e-04 0.4780336

El nombre total de gens diferencialment expressats:

```
de1 <- decideTestsDGE(et12, adjust.method="BH", p.value=0.05)
```

```
de2 <- decideTestsDGE(et13, adjust.method="BH", p.value=0.05)
```

```
de3 <- decideTestsDGE(et23, adjust.method="BH", p.value=0.05)
```

```
summary(de1);summary(de2);summary(de3)
```

2-1

Down 0

NotSig 29593

Up 0

3-1

Down 183

NotSig 27708

Up 1702

3-2

Down 0

NotSig 29593

Up 0

Clarament, les diferències més grans es troben entre els grups NIT i ELI, és a dir, entre no-infiltr. i extensament-infiltr. La funció plotSmear genera una gràfica dels canvis log-fold/log-cpm (anàlogament a una gràfica MA per a les dades de microarray):

```
par(mfrow=c(1,3))
```

```
de1tags12 <- rownames(d2)[as.logical(de1)]
```

```
plotSmear(et12, de.tags=de1tags12, main="NITvsSFI")
```

```
abline(h = c(-2, 2), col = "blue")
```

```
de1tags13 <- rownames(d2)[as.logical(de2)]
```

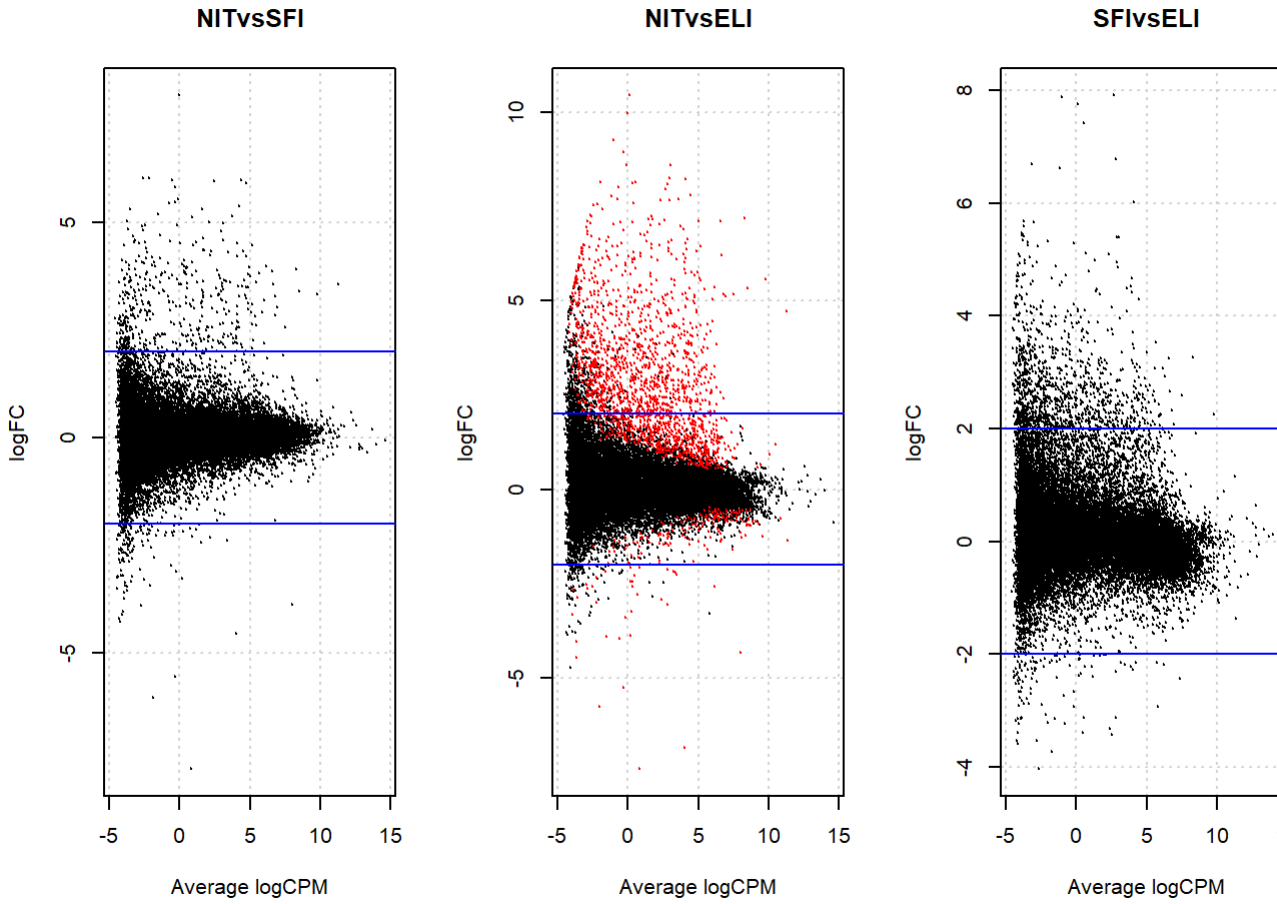
```
plotSmear(et13, de.tags=de1tags13, main="NITvsELI")
```

```
abline(h = c(-2, 2), col = "blue")
```

```
de1tags23 <- rownames(d2)[as.logical(de3)]
```

```
plotSmear(et23, de.tags=de1tags23, main="SFIvsELI")
```

```
abline(h = c(-2, 2), col = "blue")
```



4.4. GLM (model lineal generalitzat) per l´expressió diferencial.

El fem servir per trobar les etiquetes que ens interessen fent servir un criteri de verosimil.litud. Primer hem de crear una matriu de disseny per als grups. El principal supòsit aquí és que no influeix si les dades son NGS o "allele-specific expression".

gr<-as.character(group)

#Comparem únicament les diferències entres els grups d´infiltració:

design.mat<-model.matrix(~0+dgeObj$samples$group)

design.mat

dgeObj$samples$group1 dgeObj$samples$group2 dgeObj$samples$group3

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")

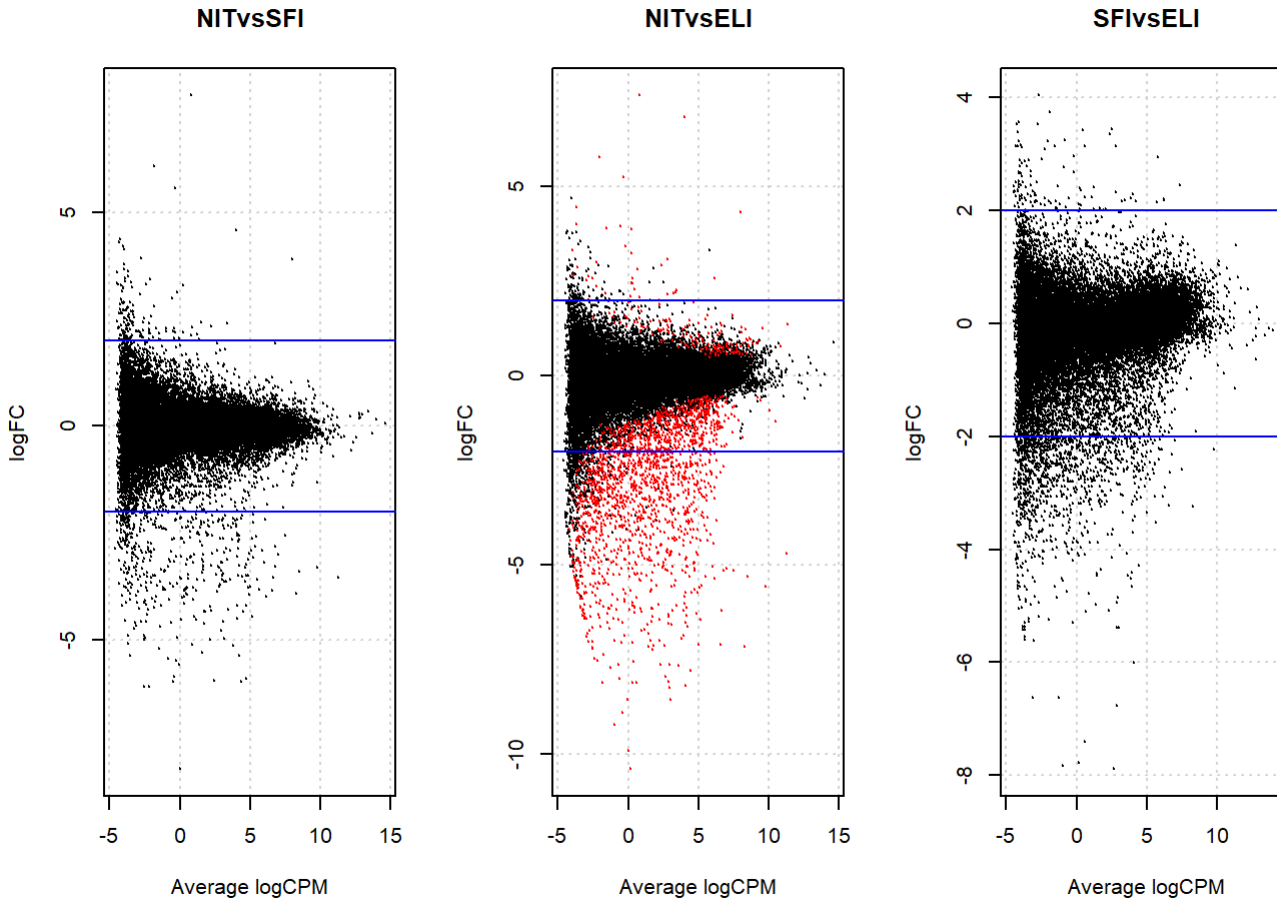
```
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$‘dgeObj$samples$group‘
## [1] "contr.treatment"
```

i ajustem el model lineal:

```
fit <- glmFit(d2, design.mat)
head(fit$coefficients)
## dgeObj$samples$group1 dgeObj$samples$group2 dgeObj$samples$group3
## 1 -16.57729 -16.69095 -17.02392
## 2 -11.18269 -11.23623 -11.18399
## 3 -17.21399 -16.80708 -17.53255
## 4 -17.26695 -17.25040 -17.88730
## 5 -15.94646 -15.78131 -15.70993
## 6 -14.91212 -14.57350 -13.63929
# si comparem (group 1 - group 2) amb 0, equival a comparar grups 1 i 2
lrt12 <- glmLRT(fit, contrast=c(1,-1,0))
lrt13 <- glmLRT(fit, contrast=c(1,0,-1))
lrt23 <- glmLRT(fit, contrast=c(0,1,-1))
topTags(lrt13, n=10)
## Coefficient: 1*dgeObj$samples$group1 -1*dgeObj$samples$group3
## logFC logCPM LR PValue FDR
## 2575 -8.199498 4.112614 82.10500 1.290459e-19 3.818855e-15
## 13513 -8.578912 3.032973 73.07064 1.250915e-17 1.850917e-13
## 26344 -6.383307 2.897115 72.18033 1.964038e-17 1.937393e-13
## 23964 -4.195523 4.030810 70.51103 4.577001e-17 2.772690e-13
## 25072 -2.787472 4.904747 70.46515 4.684705e-17 2.772690e-13
## 22733 -6.130417 4.118076 68.39069 1.341093e-16 6.614492e-13
## 2589 -3.866736 4.547101 67.77189 1.835481e-16 6.998610e-13
## 26284 -4.103448 5.579461 67.65197 1.950581e-16 6.998610e-13
## 18177 -7.008368 2.218148 67.47990 2.128459e-16 6.998610e-13
## 6871 -6.586971 3.083342 67.03655 2.665197e-16 7.887117e-13
```

i els gens més diferencialment expressats són:

```
par(mfrow=c(1,3))
dlr1 <- decideTestsDGE(lrt12, adjust.method="BH", p.value = 0.05)
dlrtags12 <- rownames(d2)[as.logical(de1)]
plotSmear(lrt12, de.tags=dlrtags12, main="NITvsSFI")
abline(h = c(-2, 2), col = "blue")
dlr2 <- decideTestsDGE(lrt13, adjust.method="BH", p.value = 0.05)
dlrtags13 <- rownames(d2)[as.logical(de2)]
plotSmear(lrt13, de.tags=dlrtags13, main="NITvsELI")
abline(h = c(-2, 2), col = "blue")
dlr3 <- decideTestsDGE(lrt23, adjust.method="BH", p.value = 0.05)
dlrtags23 <- rownames(d2)[as.logical(de3)]
plotSmear(lrt23, de.tags=dlrtags23, main="SFIvsELI")
abline(h = c(-2, 2), col = "blue")
```



Suposem ara que tenim dues variables, tipus d´infiltració i tipus mol.lecular. Només mostrarem els primers pasos d´aquest supòsit, ja que no ens interessen aquests resultats. El nostre anàlisi està basat en l´expressió diferencial entre els grups amb diferent grau d´infiltració.

fit2 <- glmFit(d2, design)

head(coef(fit2))

(Intercept) infiltrNIT infiltrSFI typeRNA Seq (NGS)

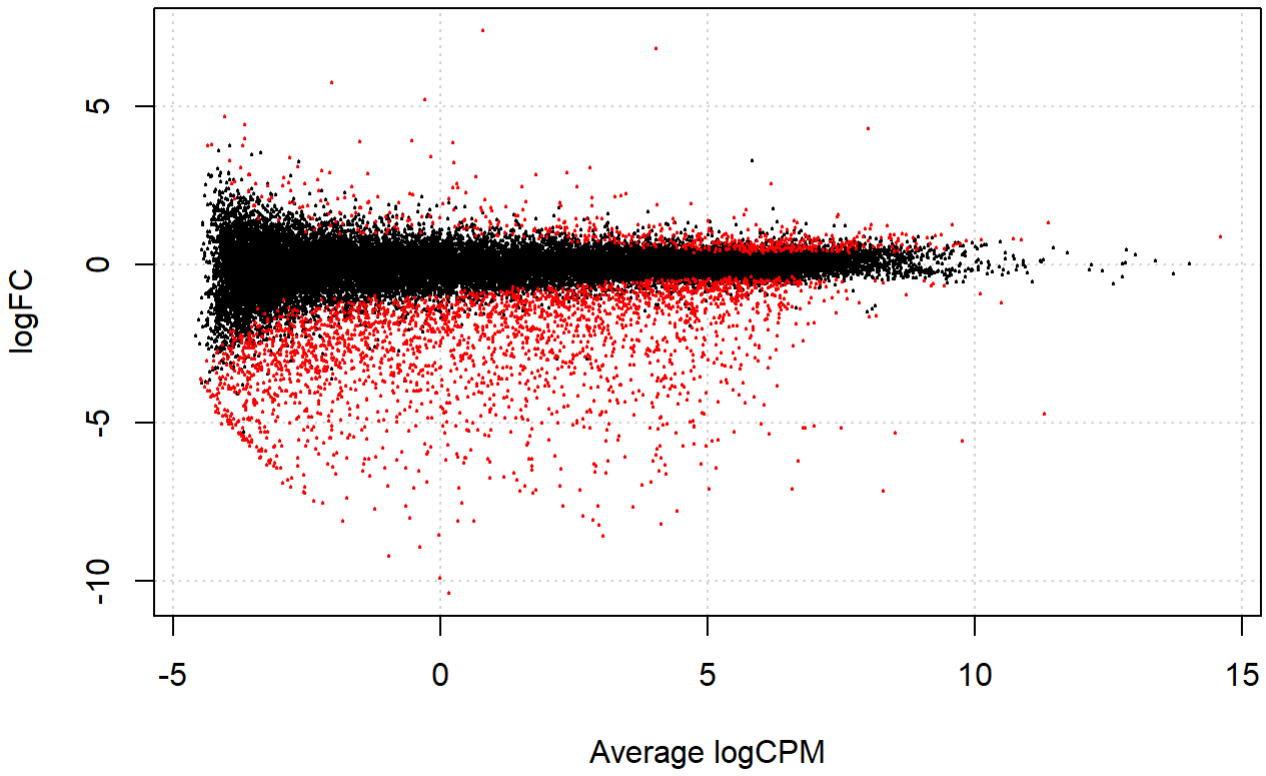
1 -16.73590 0.236834041 0.19250759 -0.375518385

2 -11.17947 -0.001385718 -0.05346605 -0.006444031

3 -17.12988 0.050332105 0.59048250 -0.648394163

4 -17.89799 0.627301215 0.63960035 0.015512222

```
## 5 -15.44154 -0.401390447 -0.11258456 -0.451326011
## 6 -13.13348 -1.675709562 -1.28161514 -0.557938339
lr <- glmLRT(fit2, coef=2)
topTags(lr)
## Coefficient: infiltrNIT
## logFC logCPM LR PValue FDR
## 2575 -8.187520 4.112614 81.84539 1.471595e-19 4.354891e-15
## 13513 -8.488006 3.032973 70.44447 4.734070e-17 7.004767e-13
## 22733 -6.241087 4.118076 67.64700 1.955502e-16 1.727876e-12
## 26344 -6.215525 2.897115 67.28321 2.351758e-16 1.727876e-12
## 25072 -2.789816 4.904747 66.85696 2.919399e-16 1.727876e-12
## 23964 -4.112775 4.030810 66.01240 4.480955e-16 2.210082e-12
## 2589 -3.846511 4.547101 65.66948 5.332532e-16 2.254366e-12
## 18177 -6.914430 2.218148 65.30000 6.432126e-16 2.379324e-12
## 23457 -4.533090 4.438357 64.96790 7.612822e-16 2.452487e-12
## 6871 -6.473664 3.083342 64.80060 8.287389e-16 2.452487e-12
results<- as.data.frame(topTags(lrt13, n=Inf))
summary(de <- decideTestsDGE(lrt13))
## 1*dgeObj$samples$group1 -1*dgeObj$samples$group3
## Down 2599
## NotSig 26419
## Up 575
summary(dlr2)
## 1*dgeObj$samples$group1 -1*dgeObj$samples$group3
## Down 2599
## NotSig 26419
## Up 575
detags <- rownames(d2)[as.logical(de)]
plotSmear(lrt13, de.tags=detags)
```



5. Annotació i visualització dels resultats.

5.1. Annotació de gens.

Ho farem mitjançant el paquet org.Mm.eg.db(), que és un dels diversos paquets *organism-level* que es reconstrueixen cada 6 mesos. Aquests paquets es mostren a la secció d'anotació de Bioconductor(http://bioconductor.org/packages/release/BiocViews.html#____AnnotationData). Un enfocament alternatiu és utilitzar biomaRt, una interfície del recurs BioMart.

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

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

```
ann[1:20,]
```

ENTREZID SYMBOL GENENAME

1 2575 <NA> <NA>

2 13513 <NA> <NA>

3 26344 <NA> <NA>

4 23964 Tenm2 teneurin transmembrane protein 2

5 25072 <NA> <NA>

6 22733 <NA> <NA>

7 2589 <NA> <NA>

8 26284 <NA> <NA>

9 18177 <NA> <NA>

10 6871 <NA> <NA>

11 25461 <NA> <NA>

12 9578 <NA> <NA>

13 12397 Cbfa2t2-ps1 CBFA2/RUNX1 translocation partner 2, pseudogene 1

14 25105 <NA> <NA>

15 21879 <NA> <NA>

16 2038 <NA> <NA>

17 23457 <NA> <NA>

18 13033 Ctsd cathepsin D

19 17952 Naip6 NLR family, apoptosis inhibitory protein 6

20 22280 <NA> <NA>

Es pot afegir l´anotació als resultats:

results.annotated <- cbind(results, ann)

results.annotated[1:25,]

logFC logCPM LR PValue FDR ENTREZID

2575 -8.199498 4.1126145 82.10500 1.290459e-19 3.818855e-15 2575

13513 -8.578912 3.0329728 73.07064 1.250915e-17 1.850917e-13 13513

26344 -6.383307 2.8971150 72.18033 1.964038e-17 1.937393e-13 26344

23964 -4.195523 4.0308098 70.51103 4.577001e-17 2.772690e-13 23964

25072 -2.787472 4.9047466 70.46515 4.684705e-17 2.772690e-13 25072

22733 -6.130417 4.1180759 68.39069 1.341093e-16 6.614492e-13 22733
2589 -3.866736 4.5471013 67.77189 1.835481e-16 6.998610e-13 2589
26284 -4.103448 5.5794613 67.65197 1.950581e-16 6.998610e-13 26284
18177 -7.008368 2.2181480 67.47990 2.128459e-16 6.998610e-13 18177
6871 -6.586971 3.0833424 67.03655 2.665197e-16 7.887117e-13 6871
25461 -6.215675 3.9670932 66.76242 3.062813e-16 8.239802e-13 25461
9578 -5.159153 3.8989572 66.50618 3.487982e-16 8.601653e-13 9578
12397 -6.092660 0.3195155 66.32885 3.816309e-16 8.687388e-13 12397
25105 -6.308875 2.8566543 65.54861 5.669812e-16 1.198477e-12 25105
21879 -7.804771 4.4162640 65.33909 6.305797e-16 1.244050e-12 21879
2038 -5.537679 2.0160205 64.80032 8.288573e-16 1.533023e-12 2038
23457 -4.407255 4.4383575 64.45552 9.873657e-16 1.718771e-12 23457
13033 -4.492374 3.9574001 62.91205 2.161456e-15 3.553553e-12 13033
17952 -3.627655 5.3227022 62.25512 3.017234e-15 4.680279e-12 17952
22280 -4.276225 4.6025309 62.16215 3.163099e-15 4.680279e-12 22280
2212 -3.916337 4.2819588 61.81159 3.779424e-15 5.325928e-12 2212
26766 -4.712149 2.0757757 60.32652 8.035758e-15 1.005668e-11 26766
5791 -6.372245 1.6956407 60.32201 8.054179e-15 1.005668e-11 5791
23878 -4.276784 1.3314173 60.29728 8.155998e-15 1.005668e-11 23878
22852 -3.290063 5.5867014 60.13022 8.878475e-15 1.050963e-11 22852
SYMBOL GENENAME
2575 <NA> <NA>
13513 <NA> <NA>
26344 <NA> <NA>
23964 Tenm2 teneurin transmembrane protein 2
25072 <NA> <NA>
22733 <NA> <NA>
2589 <NA> <NA>
26284 <NA> <NA>
18177 <NA> <NA>
6871 <NA> <NA>

25461 <NA> <NA>

9578 <NA> <NA>

12397 Cbfa2t2-ps1 CBFA2/RUNX1 translocation partner 2, pseudogene 1

25105 <NA> <NA>

21879 <NA> <NA>

2038 <NA> <NA>

23457 <NA> <NA>

13033 Ctsd cathepsin D

17952 Naip6 NLR family, apoptosis inhibitory protein 6

22280 <NA> <NA>

2212 <NA> <NA>

26766 <NA> <NA>

5791 <NA> <NA>

23878 <NA> <NA>

22852 <NA> <NA>

write.csv(results.annotated,file="NITvsELIResults.csv",row.names=FALSE)

Per decidir quins gens s'expressen de manera diferent, normalment prenem un tall de 0,05 sobre el valor p ajustat, NO el valor p brut. Això es deu al fet que estem provant més de 25000 gens i les probabilitats de trobar gens expressats de manera diferent són molt elevades quan fem moltes proves. Per tant, hem de controlar la taxa de descobriment fals(FDR), que és la columna de valor p ajustada a la taula de resultats. El que això significa és que si 100 gens són significatius a un percentatge de descobriment fals del 5%, estem disposats a acceptar que 5 seran falsos positius. La funció decideTests() mostra gens significatius al 5% de FDR.

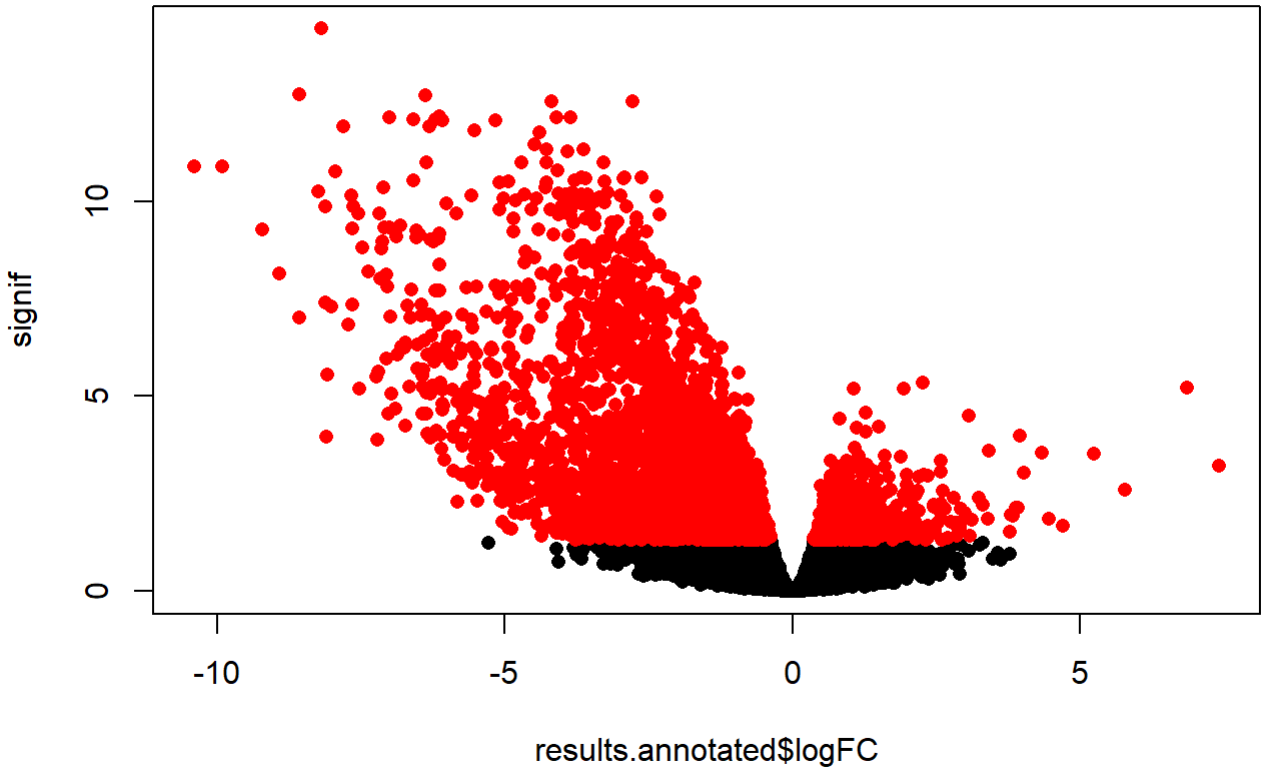
5.2. Visualització dels resultats.

Podem fer un volcano plot per veure gràficament aquests gens més significatius:

signif <- -log10(results.annotated$FDR)

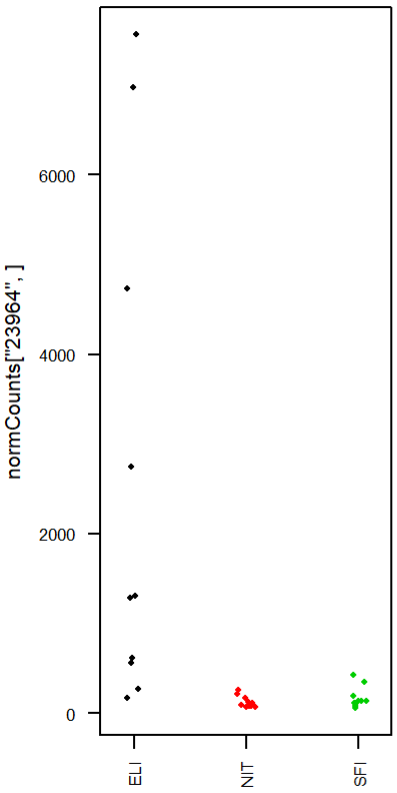
plot(results.annotated$logFC,signif,pch=16)

points(results.annotated[detags,"logFC"],-log10(results.annotated[detags,"FDR"]),pch=16,col="red")

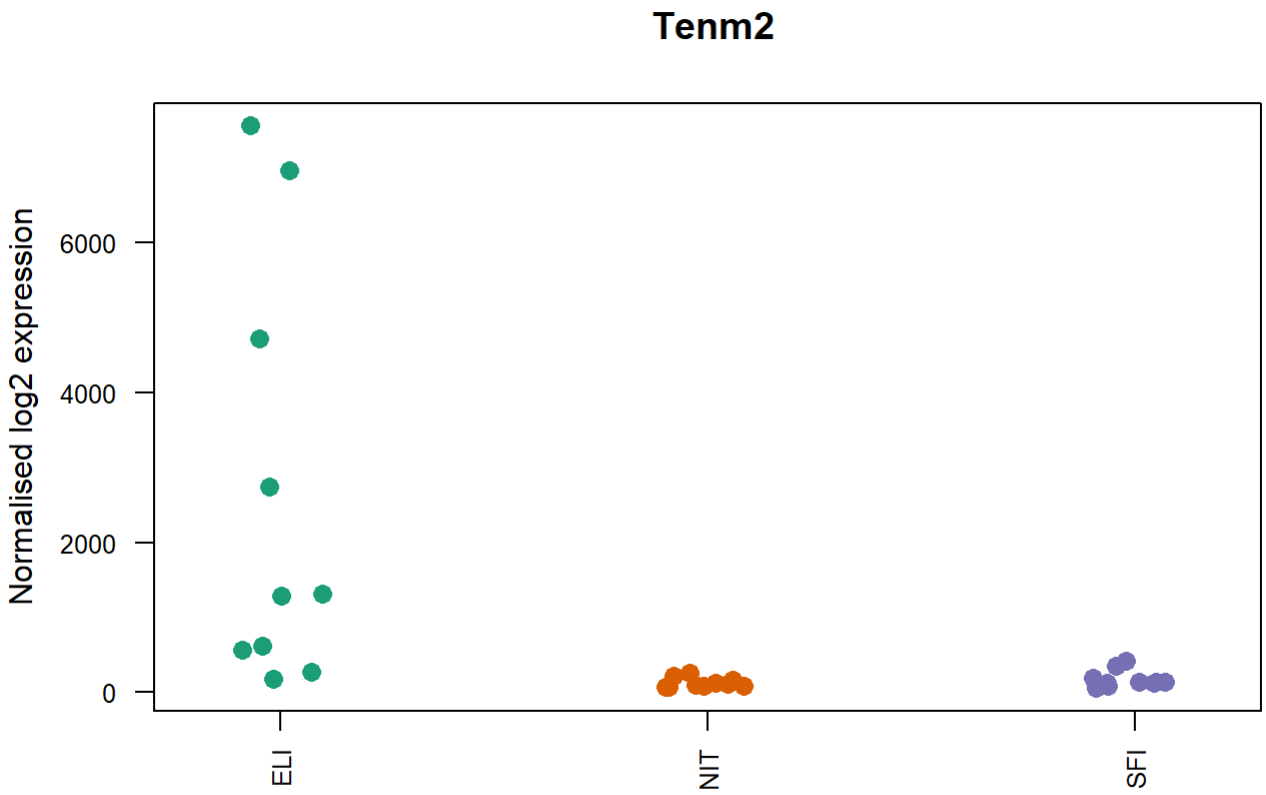


Abans de fer un seguiment dels gens diferencialment expressats amb més treballs de laboratori, es recomana fer una revisió per comprovar els nivells d'expressió de les mostres individuals per als gens d'interès. Podem mirar ràpidament l'expressió agrupada mitjançant stripchart.

```
library(RColorBrewer)

par(mfrow=c(1,3))

normCounts <- d2$counts

# Per al gen Tenm2:

stripchart(normCounts["23964",]~group,vertical=T,las=2,cex.axis=0.8,pch=16,col=1:6,method="jitter")
```




```
nice.col <- brewer.pal(6,name="Dark2")

stripchart(normCounts["23964",]~group,vertical=TRUE,las=2,cex.axis=0.8,pch=16,cex=1.3,col=nice.col,method
log2 expression",main=" Tenm2")
```



5.3. Recuperació d´ubicacions genòmiques.

Volem recuperar les coordenades dels nostres gens més significatius. Com no sabem amb quin genoma estem treballant, primer provarem amb el genoma humà hg19:

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
```

```
## Loading required package: GenomicFeatures
```

```
tx<-TxDb.Hsapiens.UCSC.hg19.knownGene
```

```
keytypes(tx)
```

```
## [1] "CDSID" "CDSNAME" "EXONID" "EXONNAME" "GENEID" "TXID"
"TXNAME"
```

```
#Prenem els 3 primers gens, Tenm2, Ctsd i Naip6:
keys <- c("23964","13033","17952")
#select(tx, keys=keys,
# keytype = "GENEID",
# columns=c("EXONCHROM","EXONSTART","EXONEND")
# )
```

No obtenim cap resultat, provem doncs amb el genoma de ratolí mm10:

```
library(TxDb.Mmusculus.UCSC.mm10.knownGene)
tx2 <- TxDb.Mmusculus.UCSC.mm10.knownGene
select(tx2, keys=keys,
keytype = "GENEID",
columns=c("EXONCHROM","EXONSTART","EXONEND")
)
## 'select()' returned 1:many mapping between keys and columns
## GENEID EXONCHROM EXONSTART EXONEND
## 1 23964 chr11 37235837 37235882
## 2 23964 chr11 37145528 37145640
## 3 23964 chr11 36943941 36944241
## 4 23964 chr11 36864668 36864943
## 5 23964 chr11 36385119 36385328
## 6 23964 chr11 36300196 36300430
## 7 23964 chr11 36273231 36273469
## 8 23964 chr11 36221408 36221530
## 9 23964 chr11 36206905 36207110
## 10 23964 chr11 36171326 36171521
## 11 23964 chr11 36163719 36163820
## 12 23964 chr11 36147015 36147209
## 13 23964 chr11 36141483 36141683
## 14 23964 chr11 36139576 36139758
## 15 23964 chr11 36106719 36106865
## 16 23964 chr11 36078360 36078570
```

```
## 17 23964 chr11 36072730 36072849
## 18 23964 chr11 36069364 36069625
## 19 23964 chr11 36068318 36068585
## 20 23964 chr11 36063777 36063920
## 21 23964 chr11 36063089 36063338
## 22 23964 chr11 36054926 36054946
## 23 23964 chr11 36051790 36052022
## 24 23964 chr11 36049092 36049246
## 25 23964 chr11 36046767 36047641
## 26 23964 chr11 36041509 36041684
## 27 23964 chr11 36039707 36039942
## 28 23964 chr11 36027145 36027441
## 29 23964 chr11 36023304 36024918
## 30 23964 chr11 36010375 36010505
## 31 23964 chr11 36006656 36008796
## 32 23964 chr11 36943941 36944166
## 33 23964 chr11 36007662 36008796
## 34 23964 chr11 36139576 36139761
## 35 23964 chr11 36008035 36008796
## 36 23964 chr11 36139480 36139761
## 37 23964 chr11 36677469 36677745
## 38 23964 chr11 36140752 36141683
## 39 23964 chr11 36494987 36495139
## 40 23964 chr11 36432143 36432276
## 41 23964 chr11 36385123 36385328
## 42 13033 chr7 142380766 142380864
## 43 13033 chr7 142377333 142377455
## 44 13033 chr7 142377026 142377170
## 45 13033 chr7 142376833 142376931
## 46 13033 chr7 142376601 142376738
## 47 13033 chr7 142371096 142371310
```

48 13033 chr7 142382592 142382667
49 13033 chr7 142380766 142380992
50 13033 chr7 142371148 142371310
51 13033 chr7 142387737 142388038
52 13033 chr7 142385459 142385618
53 13033 chr7 142383418 142383541
54 13033 chr7 142382592 142382710
55 13033 chr7 142375911 142376738
56 13033 chr7 142387737 142387857
57 13033 chr7 142377351 142377455
58 13033 chr7 142375917 142376738
59 13033 chr7 142387737 142387819
60 13033 chr7 142383194 142383541
61 17952 chr13 100315984 100316616
62 17952 chr13 100315435 100315534
63 17952 chr13 100308199 100308280
64 17952 chr13 100307027 100307078
65 17952 chr13 100304349 100304468
66 17952 chr13 100303236 100303335
67 17952 chr13 100302184 100302265
68 17952 chr13 100301335 100301392
69 17952 chr13 100298740 100300851
70 17952 chr13 100296886 100297053
71 17952 chr13 100294733 100294885
72 17952 chr13 100288101 100288187
73 17952 chr13 100285677 100285841
74 17952 chr13 100281121 100283913
75 17952 chr13 100317612 100317674
76 17952 chr13 100315984 100316675

5.4. Visió general de GenomicRanges.

GenomicRanges s´utilitza per manipular els intervals genòmics(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3738458/pdf/pcbi.1003118.pdf). Ens permet realitzar operacions comunes a intervals, com ara la superposició i el recompte. Podem definir la posició cromosòmica, inicial i final de cada regió.

```
library(GenomicRanges)

chrs <- c("chr11", "chr13","chr7")

start <- c(36000000, 100000000, 142000000)

end <- c(37300000,101000000, 143000000)

my.ranges <- GRanges(rep(chrs,3),

IRanges(start=rep(start,each=3),

end = rep(end,each=3)))

my.ranges

## GRanges object with 9 ranges and 0 metadata columns:

## seqnames ranges strand

## <Rle> <IRanges> <Rle>

## [1] chr11 36000000-37300000 *

## [2] chr13 36000000-37300000 *

## [3] chr7 36000000-37300000 *

## [4] chr11 100000000-101000000 *

## [5] chr13 100000000-101000000 *

## [6] chr7 100000000-101000000 *

## [7] chr11 142000000-143000000 *

## [8] chr13 142000000-143000000 *

## [9] chr7 142000000-143000000 *

## -------

## seqinfo: 3 sequences from an unspecified genome; no seqlengths
```

Hi ha diverses funcions útils per calcular les propietats de les dades. Més que per l'anàlisi d'ARN-seq, GenomicRanges s'utilitza a Bioconductor per a l'anàlisi de dades de NGS.

Per exemple, podem identificar regions sovint superposades entre dues GenomicRanges.

```
genePos <- select(tx2, keys=keys,
keytype = "GENEID",
columns=c("EXONCHROM","EXONSTART","EXONEND")
)
## 'select()' returned 1:many mapping between keys and columns
geneRanges <- GRanges(genePos$EXONCHROM, IRanges(genePos$EXONSTART,genePos$EXONEND),
GENEID=genePos$GENEID)
geneRanges
## GRanges object with 76 ranges and 1 metadata column:
## seqnames ranges strand | GENEID
## <Rle> <IRanges> <Rle> | <character>
## [1] chr11 37235837-37235882 * | 23964
## [2] chr11 37145528-37145640 * | 23964
## [3] chr11 36943941-36944241 * | 23964
## [4] chr11 36864668-36864943 * | 23964
## [5] chr11 36385119-36385328 * | 23964
## ... ... ... ... . ...
## [72] chr13 100288101-100288187 * | 17952
## [73] chr13 100285677-100285841 * | 17952
## [74] chr13 100281121-100283913 * | 17952
## [75] chr13 100317612-100317674 * | 17952
## [76] chr13 100315984-100316675 * | 17952
## -------
## seqinfo: 3 sequences from an unspecified genome; no seqlengths
findOverlaps(my.ranges,geneRanges)
## Hits object with 76 hits and 0 metadata columns:
## queryHits subjectHits
## <integer> <integer>
## [1] 1 1
## [2] 1 2
## [3] 1 3
```

[4] 1 4

[5] 1 5

... ...

[72] 9 56

[73] 9 57

[74] 9 58

[75] 9 59

[76] 9 60

----

queryLength: 9 / subjectLength: 76

seqlevelsStyle(geneRanges)

[1] "UCSC"

Recuperació de les coordenades genètiques com a GenomicRanges.

La sortida de exonsBy és una llista d´exons:

exo <- exonsBy(tx2,"gene")

exo

GRangesList object of length 24594:

$‘100009600‘

GRanges object with 9 ranges and 2 metadata columns:

seqnames ranges strand | exon_id exon_name

<Rle> <IRanges> <Rle> | <integer> <character>

[1] chr9 21062393-21062717 - | 233853 <NA>

[2] chr9 21062400-21062717 - | 233855 <NA>

[3] chr9 21062894-21062987 - | 233856 <NA>

[4] chr9 21063314-21063396 - | 233857 <NA>

[5] chr9 21066024-21066377 - | 233858 <NA>

[6] chr9 21066940-21067093 - | 233859 <NA>

[7] chr9 21066940-21067925 - | 233860 <NA>

[8] chr9 21068030-21068117 - | 233867 <NA>

[9] chr9 21073075-21073096 - | 233869 <NA>

```
## -------
## seqinfo: 66 sequences (1 circular) from mm10 genome
##
## $`100009609`
## GRanges object with 8 ranges and 2 metadata columns:
## seqnames ranges strand | exon_id exon_name
## <Rle> <IRanges> <Rle> | <integer> <character>
## [1] chr7 84935565-84941088 - | 190288 <NA>
## [2] chr7 84943141-84943264 - | 190289 <NA>
## [3] chr7 84943504-84943722 - | 190290 <NA>
## [4] chr7 84943504-84947000 - | 190291 <NA>
## [5] chr7 84946200-84947000 - | 190292 <NA>
## [6] chr7 84947372-84947651 - | 190293 <NA>
## [7] chr7 84948507-84949184 - | 190294 <NA>
## [8] chr7 84963816-84964115 - | 190295 <NA>
## -------
## seqinfo: 66 sequences (1 circular) from mm10 genome
##
## $`100009614`
## GRanges object with 1 range and 2 metadata columns:
## seqnames ranges strand | exon_id exon_name
## <Rle> <IRanges> <Rle> | <integer> <character>
## [1] chr10 77711457-77712009 + | 250227 <NA>
## -------
## seqinfo: 66 sequences (1 circular) from mm10 genome
##
## ...
## <24591 more elements>
```

Per accedir a l'estructura d'un determinat gen, podem utilitzar la sintaxi [[]] amb el nom del gen (Entrez gen ID) entre cometes. Si volguéssim una regió sencera que abasta el gen, podríem utilitzar la funció range

range(exo[["23964"]])

GRanges object with 1 range and 0 metadata columns:

seqnames ranges strand

<Rle> <IRanges> <Rle>

[1] chr11 36006656-37235882 -

----

seqinfo: 66 sequences (1 circular) from mm10 genome

6. Exportació de pistes.

També és possible guardar els resultats d'una anàlisi de bioconductors en un navegador per permetre l'anàlisi i la integració interactives amb altres tipus de dades o compartir amb col · laboradors. Per exemple, potser voldríem que una pista del navegador indiqui on es troben els nostres gens expressats de manera diferent. Utilitzarem el bedformat per mostrar aquestes ubicacions. Anotarem els intervals amb informació de la nostra anàlisi, com ara el canvi de plec i la importància.

Primer creem un marc de dades només per als gens DE.

sigGenes <- results.annotated[detags,]

sigGenes[1:15,]

logFC logCPM LR PValue FDR ENTREZID SYMBOL

22 4.3301590 7.98272330 19.443684 1.036099e-05 2.914570e-04 22 <NA>

40 -2.5466636 0.33483956 17.866363 2.369729e-05 5.968288e-04 40 <NA>

44 -0.6657930 4.23499016 8.589671 3.380751e-03 3.524007e-02 44 <NA>

45 -1.6686439 1.12764638 15.370562 8.835412e-05 1.829716e-03 45 <NA>

46 -1.3137211 -0.06459686 13.559471 2.311222e-04 4.085782e-03 46 <NA>

57 -2.5182961 -2.06396284 23.719925 1.114230e-06 4.390598e-05 57 <NA>

62 -3.2719597 1.56920004 33.647753 6.605114e-09 5.282841e-07 62 <NA>

76 -1.5843594 2.04669876 14.565905 1.353414e-04 2.624611e-03 76 <NA>

89 0.9688761 6.81421533 10.428188 1.241064e-03 1.622926e-02 89 <NA>

137 1.5519076 -1.67692763 8.322496 3.915709e-03 3.937396e-02 137 <NA>

140 -0.9285109 0.95312010 8.662042 3.249070e-03 3.419265e-02 140 <NA>

143 -1.1865915 -1.52266034 9.264939 2.335823e-03 2.647415e-02 143 <NA>

144 -1.8568946 -0.59065283 24.615599 6.998295e-07 2.941769e-05 144 <NA>

150 0.5779696 5.90320119 9.216798 2.398046e-03 2.698984e-02 150 <NA>

153 0.5209950 5.14762323 9.307532 2.282137e-03 2.601513e-02 153 <NA>

GENENAME

22 <NA>

40 <NA>

44 <NA>

45 <NA>

46 <NA>

57 <NA>

62 <NA>

76 <NA>

89 <NA>

137 <NA>

140 <NA>

143 <NA>

144 <NA>

150 <NA>

153 <NA>

Fem la funció range() per obtenir un únic rang per a cada gen i tranformar-lo a un objecte més convenient.

exoRanges <- unlist(range(exo))

sigRegions <- exoRanges[na.omit(match(sigGenes$ENTREZID, names(exoRanges)))]

sigRegions

GRanges object with 641 ranges and 0 metadata columns:

seqnames ranges strand

<Rle> <IRanges> <Rle>

11302 chr11 120007313-120047167 -

11409 chr5 115110299-115119346 -

11438 chr2 181018380-181043546 -

11535 chr7 110627661-110629820 +

11545 chr1 180568924-180601254 +

... ...

28078 chr13 28142484-28151611 +

28105 chr18 46165300-46212607 -

28248 chr6 141907282-141946962 -

28250 chr6 141805440-141856199 -

28254 chr6 142085761-142208521 -

----

seqinfo: 66 sequences (1 circular) from mm10 genome

En lloc de només representar les ubicacions genòmiques, el format .bed també és capaç de acolorir cada rang d'acord amb alguna propietat de l'anàlisi (per exemple, la direcció i la magnitud del canvi) per ajudar a ressaltar determinades regions d'interès. També es pot mostrar una puntuació quan es fa clic en una regió. Una utilitat útil de GenomicRanges és que podem adjuntar metadades a cada rang mitjançant la funció mcols. Les metadades es poden subministrar en forma de marc de dades.

mcols(sigRegions) <- sigGenes[match(names(sigRegions), rownames(sigGenes)),]

sigRegions

GRanges object with 641 ranges and 8 metadata columns:

seqnames ranges strand | logFC

<Rle> <IRanges> <Rle> | <numeric>

11302 chr11 120007313-120047167 - | 0.817429530179723

11409 chr5 115110299-115119346 - | -0.669098428324545

11438 chr2 181018380-181043546 - | 0.910920109606485

11535 chr7 110627661-110629820 + | -1.65442398063402

11545 chr1 180568924-180601254 + | 0.541591455513655

... ...

28078 chr13 28142484-28151611 + | -0.548374279995536

28105 chr18 46165300-46212607 - | -0.790435228449426

28248 chr6 141907282-141946962 - | -3.17291659902311

28250 chr6 141805440-141856199 - | -1.30387240129358

28254 chr6 142085761-142208521 - | -2.58934287496386

logCPM LR PValue

```
## <numeric> <numeric> <numeric>
## 11302 7.73127609737513 15.8598193311691 6.82116735336549e-05
## 11409 4.2841609014407 12.3999752293265 0.000429339544166249
## 11438 2.6896442856872 11.2400510640191 0.000800509536564958
## 11535 0.339512516871309 18.4170097427836 1.77466711213033e-05
## 11545 6.44432199378929 11.9403791355403 0.000549303003030274
## ... ... ... ...
## 28078 4.92068805235374 9.41405383750674 0.00215328643143392
## 28105 3.00021111478038 9.47167004497339 0.00208669318307512
## 28248 3.60713480367163 49.4755211271024 2.00862442262342e-12
## 28250 -1.91584362715332 8.22368416414385 0.00413471741117757
## 28254 -0.449062990216679 16.6686515539186 4.4510492050657e-05
## FDR ENTREZID SYMBOL
## <numeric> <character> <character>
## 11302 0.00147342193786967 11302 Aatk
## 11409 0.00680527323541072 11409 Acads
## 11438 0.0113385291413015 11438 Chrna4
## 11535 0.000462711223341612 11535 Adm
## 11545 0.00838778316237095 11545 Parp1
## ... ... ... ...
## 28078 0.0249304402838122 28078 Prl5a1
## 28105 0.0243780911842719 28105 Trim36
## 28248 5.94412225386947e-10 28248 Slco1a1
## 28250 0.041101341064487 28250 Slco1a4
## 28254 0.00102745631143143 28254 Slco1a6
## GENENAME
## <character>
## 11302 apoptosis-associated tyrosine kinase
## 11409 acyl-Coenzyme A dehydrogenase, short chain
## 11438 cholinergic receptor, nicotinic, alpha polypeptide 4
## 11535 adrenomedullin
```

11545 poly (ADP-ribose) polymerase family, member 1

... ...

28078 prolactin family 5, subfamily a, member 1

28105 tripartite motif-containing 36

28248 solute carrier organic anion transporter family, member 1a1

28250 solute carrier organic anion transporter family, member 1a4

28254 solute carrier organic anion transporter family, member 1a6

----

seqinfo: 66 sequences (1 circular) from mm10 genome

Les metadades que hem afegit també es poden utilitzar com a mitjà per interrogar els intervals; com si les dades estiguessin contingudes en un marc de dades.

sigRegions[order(sigRegions$LR,decreasing = TRUE)]

GRanges object with 641 ranges and 8 metadata columns:

seqnames ranges strand | logFC

<Rle> <IRanges> <Rle> | <numeric>

23964 chr11 36006656-37235882 - | -4.19552267953141

13033 chr7 142371096-142388038 - | -4.4923740407502

17952 chr13 100281121-100317674 - | -3.627654525912

17883 chr11 67078300-67102291 + | -2.62239151914312

21935 chr16 11313812-11320074 + | -3.2604938601378

... ...

14396 chrX 72432681-72656848 - | 2.31077325179895

22635 chr5 137378637-137477064 - | 0.694257681183349

20130 chr7 45017961-45021647 + | 0.450382433245389

17973 chr9 100492293-100546134 - | 1.15038620980898

18811 chr13 12996125-13005383 - | -0.651711075221602

logCPM LR PValue

<numeric> <numeric> <numeric>

23964 4.03080980026957 70.5110305598534 4.57700113607097e-17

13033 3.95740011535145 62.9120520655987 2.16145578444457e-15

17952 5.3227021642845 62.2551206937359 3.01723372267764e-15

```
## 17883 5.94750384318781 58.1055531821216 2.48424050855975e-14
## 21935 6.12066328355485 57.0916375358171 4.1597066494777e-14
## ... ... ... ...
## 14396 -3.596347796917 7.80792632527471 0.00520175609985951
## 22635 4.74883797117765 7.80063882850468 0.00522277663812322
## 20130 6.9381408538552 7.78590167993713 0.00526555043557708
## 17973 2.87831840816432 7.77979012393102 0.00528339349625587
## 18811 5.10877213262751 7.77798271321455 0.00528868214227241
## FDR ENTREZID SYMBOL
## <numeric> <character> <character>
## 23964 2.77268977201047e-13 23964 Tenm2
## 13033 3.55355339050379e-12 13033 Ctsd
## 17952 4.68027948789732e-12 17952 Naip6
## 17883 2.41522122565773e-11 17883 Myh3
## 21935 3.23942628626299e-11 21935 Tnfrsf17
## ... ... ... ...
## 14396 0.0488229373254622 14396 Gabra3
## 22635 0.0489881550085516 22635 Zan
## 20130 0.0492956134261413 20130 Rras
## 17973 0.0494230395423716 17973 Nck1
## 18811 0.0494339768276271 18811 Prl2c2
## GENENAME
## <character>
## 23964 teneurin transmembrane protein 2
## 13033 cathepsin D
## 17952 NLR family, apoptosis inhibitory protein 6
## 17883 myosin, heavy polypeptide 3, skeletal muscle, embryonic
## 21935 tumor necrosis factor receptor superfamily, member 17
## ... ...
## 14396 gamma-aminobutyric acid (GABA) A receptor, subunit alpha 3
## 22635 zonadhesin
```

20130 related RAS viral (r-ras) oncogene

17973 non-catalytic region of tyrosine kinase adaptor protein 1

18811 prolactin family 2, subfamily c, member 2

----

seqinfo: 66 sequences (1 circular) from mm10 genome

Score <- -log10(sigRegions$FDR)

Ara crearem una puntuació a partir dels valors p que es mostraran a cada regió i l'esquema de colors per a les regions en funció del *fold-change*. Per a la puntuació podem utilitzar el \(-log_{10}\) del valor p ajustat. A l'hora d'escollir paletes de colors, ens podem preocupar de que siguin adequades per a daltònics . L'esquema de colors vermell/verd aplicat tradicionalment als microarrays és una *mala* elecció.

rbPal <-colorRampPalette(c("red", "blue"))

logfc <- pmax(sigRegions$logFC, -5)

logfc <- pmin(logfc , 5)

Col <- rbPal(10)[as.numeric(cut(logfc, breaks = 10))]

Els colors i la puntuació s'han de desar a GRanges com a score i itemRgb respectivament, i s'utilitzaran per construir la pista del navegador. Ara podem exportar els resultats significatius de l'anàlisi d´expressió diferencial com a .bed utilitzant rtracklayer. Es pot carregar el fitxer resultant a IGV.

mcols(sigRegions)$score <- Score

mcols(sigRegions)$itemRgb <- Col

library(rtracklayer)

export(sigRegions , con = "topHits.bed")

7. Proves de significació biològica competitives.

7.1. Anàlisi GOseq.

GOseq és un mètode per realitzar un anàlisi ontològic de gens, adequat per a dades de RNA-seq, ja que dóna compte del biaix de longitud del gen en la detecció de sobrerepresentació.

Primerament, GOseq ha de quantificar el biaix de longitud present al conjunt de dades a considerar. Això es fa calculant una funció de ponderació de probabilitats o PWF, que es pot pensar com una funció que dóna la probabilitat que un gen s'expressi diferencialment (DE), basat només en la seva longitud. El PWF es calcula ajustant una spline monotònica a la sèrie de dades binàries d'expressió

diferencial (1 = DE, 0 = Not DE) en funció de la longitud del gen. El PWF s'utilitza per ponderar la possibilitat de seleccionar cada gen quan es formi una distribució nul · la per a membres de la categoria GO. El fet que el PWF es calculi directament a partir del conjunt de dades analitzat fa que aquest plantejament sigui robust, només corregint el biaix de longitud present a les dades. "L'anàlisi de GO de dades de RNA-seq requereix l'ús de mostreig aleatori per generar una distribució nula adequada per a membres de la categoria GO i calcular la importància de cada categoria per a la sobrerepresentació entre els gens DE. En la majoria dels casos, la distribució de Wallenius es pot utilitzar per aproximar la distribució nul real, sense cap pèrdua significativa de precisió. El paquet goseq implementa aquesta aproximació com a opció predeterminada." (GOseq vignette)

Creem una llista de DEG:

```
print(head(results))
```

```
## logFC logCPM LR PValue FDR
## 2575 -8.199498 4.112614 82.10500 1.290459e-19 3.818855e-15
## 13513 -8.578912 3.032973 73.07064 1.250915e-17 1.850917e-13
## 26344 -6.383307 2.897115 72.18033 1.964038e-17 1.937393e-13
## 23964 -4.195523 4.030810 70.51103 4.577001e-17 2.772690e-13
## 25072 -2.787472 4.904747 70.46515 4.684705e-17 2.772690e-13
## 22733 -6.130417 4.118076 68.39069 1.341093e-16 6.614492e-13
```

```
genes <- results$FDR < 0.01
names(genes) <- rownames(results)
print(head(genes))
```

```
## 2575 13513 26344 23964 25072 22733
## TRUE TRUE TRUE TRUE TRUE TRUE
```

Ajustem la PWF:

```
library(goseq)
```

```
## Loading required package: BiasedUrn
## Loading required package: geneLenDataBase
##
## Attaching package: 'geneLenDataBase'
## The following object is masked from 'package:S4Vectors':
##
```

unfactor

##

#print(supportedGeneIDs())

#print(supportedGenomes())

pwf <- nullp(genes, "mm10","knownGene")

Can't find mm10/knownGene length data in genLenDataBase...

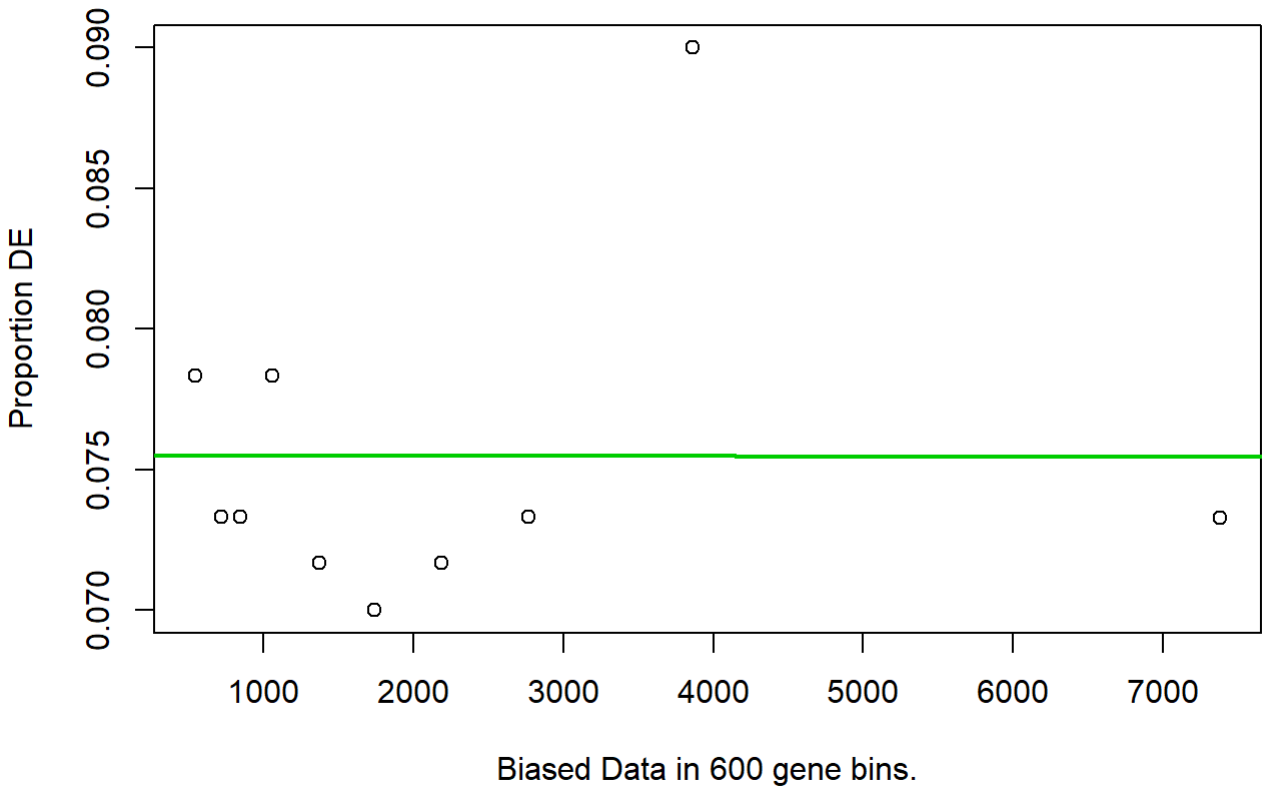
Found the annotation package, TxDb.Mmusculus.UCSC.mm10.knownGene

Trying to get the gene lengths from it.

Warning in getlength(names(DEgenes), genome, id): More than 40% of gene names specified did not match the gene names for genome mm10 and ID knownGene. No length data will be available for these genes.

Gene names which failed to match were: 2575, 13513, 26344, 25072, 22733, 2589, 26284, 18177, 6871, 25461

Required gene names are: 100009600, 100009609, 100009614, 100009664, 100012, 100017, 100019, 100033459, 100034251, 100034361



Fem l´anàlisis d'enriquiment (*Gen Enrichment Analysis*):

go.results <- goseq(pwf, "mm10","knownGene")

Fetching GO annotations...

Warning in goseq(pwf, "mm10", "knownGene"): Missing length data for 81% of

genes. Accuarcy of GO test will be reduced.

For 23861 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 documentation).

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
go.results[1:15,]
## category over_represented_pvalue under_represented_pvalue numDEInCat
## 1857 GO:0004725 1.870863e-10 1.0000000 20
## 1853 GO:0004721 1.332808e-07 1.0000000 20
## 5193 GO:0016791 1.915358e-06 0.9999995 22
## 2722 GO:0006470 9.026447e-06 0.9999975 22
## 357 GO:0001518 1.205973e-05 0.9999996 6
## 8134 GO:0034706 1.269845e-05 0.9999990 8
## 4978 GO:0016311 1.613900e-05 0.9999948 25
## 2194 GO:0005381 3.156255e-05 1.0000000 4
## 8315 GO:0035335 8.147587e-05 0.9999928 7
## 9219 GO:0042578 2.133884e-04 0.9999218 23
## 17735 GO:1904949 2.401036e-04 0.9999667 8
## 5667 GO:0019829 2.427022e-04 0.9999603 9
## 6074 GO:0022853 2.427022e-04 0.9999603 9
## 9244 GO:0042625 2.427022e-04 0.9999603 9
## 2147 GO:0005248 3.110869e-04 0.9999721 6
## numInCat term ontology
## 1857 51 protein tyrosine phosphatase activity MF
## 1853 71 phosphoprotein phosphatase activity MF
## 5193 97 phosphatase activity MF
## 2722 106 protein dephosphorylation BP
## 357 9 voltage-gated sodium channel complex CC
## 8134 17 sodium channel complex CC
## 4978 134 dephosphorylation BP
## 2194 4 iron ion transmembrane transporter activity MF
## 8315 16 peptidyl-tyrosine dephosphorylation BP
## 9219 138 phosphoric ester hydrolase activity MF
## 17735 24 ATPase complex CC
```

5667 30 cation-transporting ATPase activity MF

6074 30 active ion transmembrane transporter activity MF

9244 30 ATPase coupled ion transmembrane transporter activity MF

2147 14 voltage-gated sodium channel activity MF

7.2. FGSEA.

L'anàlisi d'enriquiment ràpid de conjunts genètics pre-consultats (GSEA) es realitza:

- classificant tots els gens del conjunt de dades segons la seva correlació amb el fenotip escollit,
- identificant les posicions de rang de tots els membres del conjunt de gens i
- calculant una puntuació d'enriquiment (ES) que representa la diferència entre els rànquings observats i el que s'esperaria suposant una distribució de rànquing aleatori.

"Després d'establir l'ES per a cada grup de gens a través del fenotip, GSEA reiteradament randomitza les etiquetes de les mostres i torna a fer l'enriquiment a través de les classes aleatòries. Mitjançant repetides randomitzacions d'etiquetes de classe, es pot comparar l'ES de cada conjunt de gens a través de les classes veritables amb la distribució ES de les classes aleatòries. Es consideren significatius aquells conjunts de gens que superen significativament les permutacions iteratives de classe aleatòria." De fgsea[vignette] (http://www.bioconductor.org/packages/release/bioc/vignettes/fgsea/inst/doc/fgsea-tutorial.html) "fast preranked gene set enrichment analysis (GSEA)".

library(fgsea)

Loading required package: Rcpp

results.ord <- results[order(-results[,"logFC"]),]

head(results.ord)

logFC logCPM LR PValue FDR

5759 7.408705 0.7784657 17.778632 2.481546e-05 6.186721e-04

15835 6.847999 4.0177782 28.177392 1.106900e-07 6.037119e-06

24728 5.779873 -2.0480865 14.671822 1.279448e-04 2.512455e-03

23602 5.237948 -0.3027408 19.393131 1.063890e-05 2.984238e-04

27352 4.690643 -4.0458205 9.717483 1.825233e-03 2.184154e-02

20962 4.456888 -3.6843629 10.696537 1.073362e-03 1.438588e-02

ranks <- results.ord$logFC

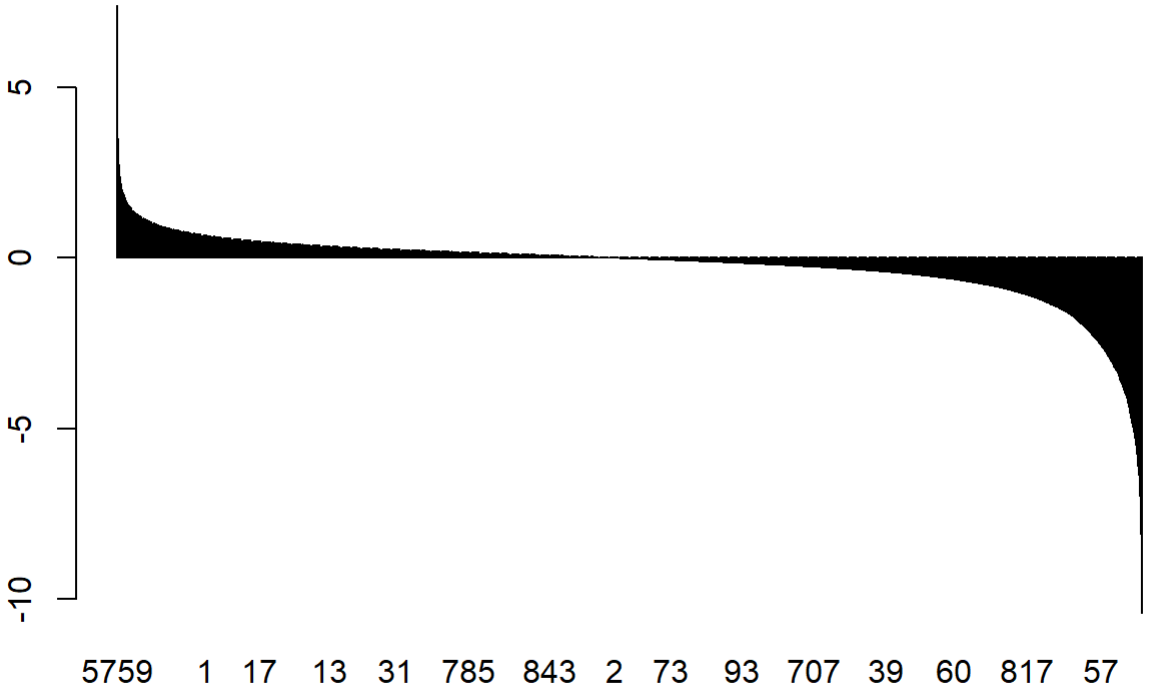
names(ranks) <- rownames(results.ord)

head(ranks)

5759 15835 24728 23602 27352 20962

7.408705 6.847999 5.779873 5.237948 4.690643 4.456888

barplot(ranks)



Carreguem les rutes:

load("mouse_H_v5.rdata")

pathways <- Mm.H

Anàlisi de conducta:

fgseaRes <- fgsea(pathways, ranks, minSize=15, maxSize = 500, nperm=1000)

fgseaRes[order(padj),]

pathway pval padj ES

1: HALLMARK_CHOLESTEROL_HOMEOSTASIS 0.004439512 0.1370614 -0.6790700

2: HALLMARK_UNFOLDED_PROTEIN_RESPONSE 0.005482456 0.1370614 -0.6613743

3: HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION 0.010050251 0.1588983 -0.5115251

4: HALLMARK_BILE_ACID_METABOLISM 0.012711864 0.1588983 -0.5808456

5: HALLMARK_APICAL_JUNCTION 0.116044400 0.5826626 -0.4683589

6: HALLMARK_COMPLEMENT 0.121334681 0.5826626 -0.4574610

7: HALLMARK_MTORC1_SIGNALING 0.121827411 0.5826626 -0.4696151

8: HALLMARK_MYC_TARGETS_V2 0.163145540 0.5826626 -0.5625417

9: HALLMARK_P53_PATHWAY 0.065989848 0.5826626 -0.4912743

10: HALLMARK_UV_RESPONSE_DN 0.097959184 0.5826626 -0.4875485

11: HALLMARK_PEROXISOME 0.149681529 0.5826626 -0.4961710

12: HALLMARK_SPERMATOGENESIS 0.145194274 0.5826626 -0.4721273

13: HALLMARK_KRAS_SIGNALING_DN 0.099694812 0.5826626 -0.4781859

14: HALLMARK_PANCREAS_BETA_CELLS 0.159502262 0.5826626 -0.5476568

15: HALLMARK_IL6_JAK_STAT3_SIGNALING 0.209812109 0.6170944 -0.4668373

16: HALLMARK_APOPTOSIS 0.198588710 0.6170944 -0.4435857

17: HALLMARK_COAGULATION 0.201424212 0.6170944 -0.4547494

18: HALLMARK_DNA_REPAIR 0.300847458 0.6912975 -0.4485570

19: HALLMARK_PROTEIN_SECRETION 0.287356322 0.6912975 0.3020081

20: HALLMARK_HEDGEHOG_SIGNALING 0.302036199 0.6912975 -0.4846936

21: HALLMARK_UV_RESPONSE_UP 0.304170905 0.6912975 -0.4260524

22: HALLMARK_KRAS_SIGNALING_UP 0.281533804 0.6912975 -0.4264843

23: HALLMARK_IL2_STAT5_SIGNALING 0.320925553 0.6976642 -0.4142598

24: HALLMARK_HYPOXIA 0.335010060 0.6979376 -0.4126026

25: HALLMARK_HEME_METABOLISM 0.402451481 0.8049030 -0.4107206

26: HALLMARK_TNFA_SIGNALING_VIA_NFKB 0.466331658 0.8967917 -0.3900719

27: HALLMARK_MITOTIC_SPINDLE 0.582815735 0.9114959 -0.3755889

28: HALLMARK_TGF_BETA_SIGNALING 0.641111111 0.9114959 -0.3784488

29: HALLMARK_ADIPOGENESIS 0.736788618 0.9114959 -0.3393259

30: HALLMARK_ESTROGEN_RESPONSE_EARLY 0.725330621 0.9114959 -0.3443260

31: HALLMARK_ESTROGEN_RESPONSE_LATE 0.765656566 0.9114959 -0.3326678

32: HALLMARK_ANDROGEN_RESPONSE 0.606382979 0.9114959 -0.3814914

33: HALLMARK_INTERFERON_GAMMA_RESPONSE 0.765419616 0.9114959 -0.3321834

34: HALLMARK_PI3K_AKT_MTOR_SIGNALING 0.741631799 0.9114959 -0.3397783

35: HALLMARK_MYC_TARGETS_V1 0.732723577 0.9114959 -0.3419050

36: HALLMARK_INFLAMMATORY_RESPONSE 0.494461229 0.9114959 -0.3866170

37: HALLMARK_XENOBIOTIC_METABOLISM 0.531504065 0.9114959 -0.3821830

38: HALLMARK_FATTY_ACID_METABOLISM 0.648373984 0.9114959 -0.3590487

39: HALLMARK_OXIDATIVE_PHOSPHORYLATION 0.692546584 0.9114959 -0.3516907

40: HALLMARK_GLYCOLYSIS 0.650406504 0.9114959 -0.3585292

41: HALLMARK_ANGIOGENESIS 0.643348624 0.9114959 -0.3801679

42: HALLMARK_ALLOGRAFT_REJECTION 0.671342685 0.9114959 -0.3546446

43: HALLMARK_MYOGENESIS 0.822110553 0.9134562 -0.3197404

44: HALLMARK_APICAL_SURFACE 0.812500000 0.9134562 -0.3391265

45: HALLMARK_REACTIVE_OXIGEN_SPECIES_PATHWAY 0.801643192 0.9134562 -0.3285954

46: HALLMARK_NOTCH_SIGNALING 0.897849462 0.9664293 0.2466021

47: HALLMARK_E2F_TARGETS 0.908443540 0.9664293 -0.2853510

48: HALLMARK_G2M_CHECKPOINT 0.951564077 0.9709838 -0.2533312

49: HALLMARK_INTERFERON_ALPHA_RESPONSE 0.935911602 0.9709838 -0.2619305

50: HALLMARK_WNT_BETA_CATENIN_SIGNALING 0.977728285 0.9777283 -0.2195892

pathway pval padj ES

NES nMoreExtreme size leadingEdge

1: -1.6267579 3 39 19016,20249,20239,27279,11910,12660,...

2: -1.6101861 4 44 27966,18044,22027,11911,18045,11910,...

3: -1.3467971 9 158 18645,20229,22003,20254,16774,12505,...

4: -1.4495633 11 64 20181,20183,20238,20186,12359,11611,...

5: -1.2208231 114 129 16151,16774,18175,19261,19264,12308,...

6: -1.1976497 119 144 20255,13033,20196,16993,13482,12902,...

7: -1.2205290 119 120 27966,20249,12035,22042,18046,22027,...

8: -1.2816350 138 27 27966,27993,18673,18148

9: -1.2768210 64 120 20181,20194,20229,20198,13033,19267,...

10: -1.2577452 95 96 19016,20181,20238,20273,21813,19274,...

11: -1.2355676 140 67 20183,20238,12359,12613,15488,13370,...

12: -1.2151089 141 94 20255,22024,18168,16162,20394,13012,...

13: -1.2414166 97 116 20191,20264,20190,23964,14760,22094,...

14: -1.2823924 140 32 13482,20604,18508,23797,11925,18088,...

15: -1.1778448 200 75 15975,12505,27279,16161,19247,19246,...

16: -1.1583462 196 134 20229,12505,27279,22035,18646,20230,...

17: -1.1777539 197 104 22041,20196,20193,16993,13482,13034,...

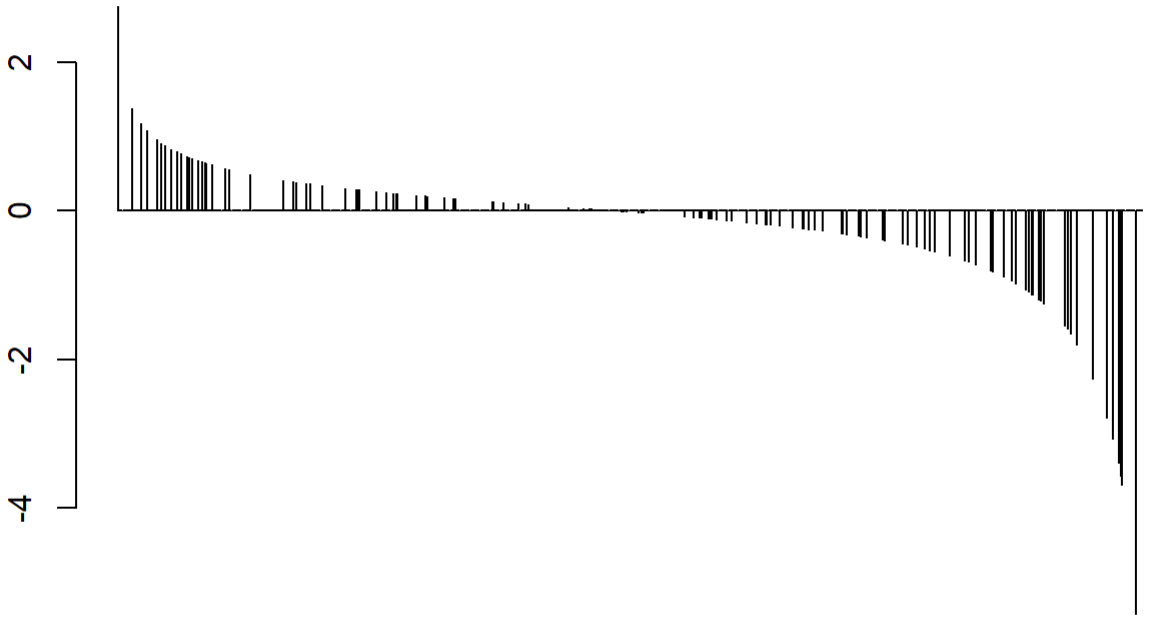
18: -1.1194228 283 64 20020,27369,20018,13716,26370,13872,...

19: 1.0935752 24 45 11303,14420,12512,11773,18484,12757,...

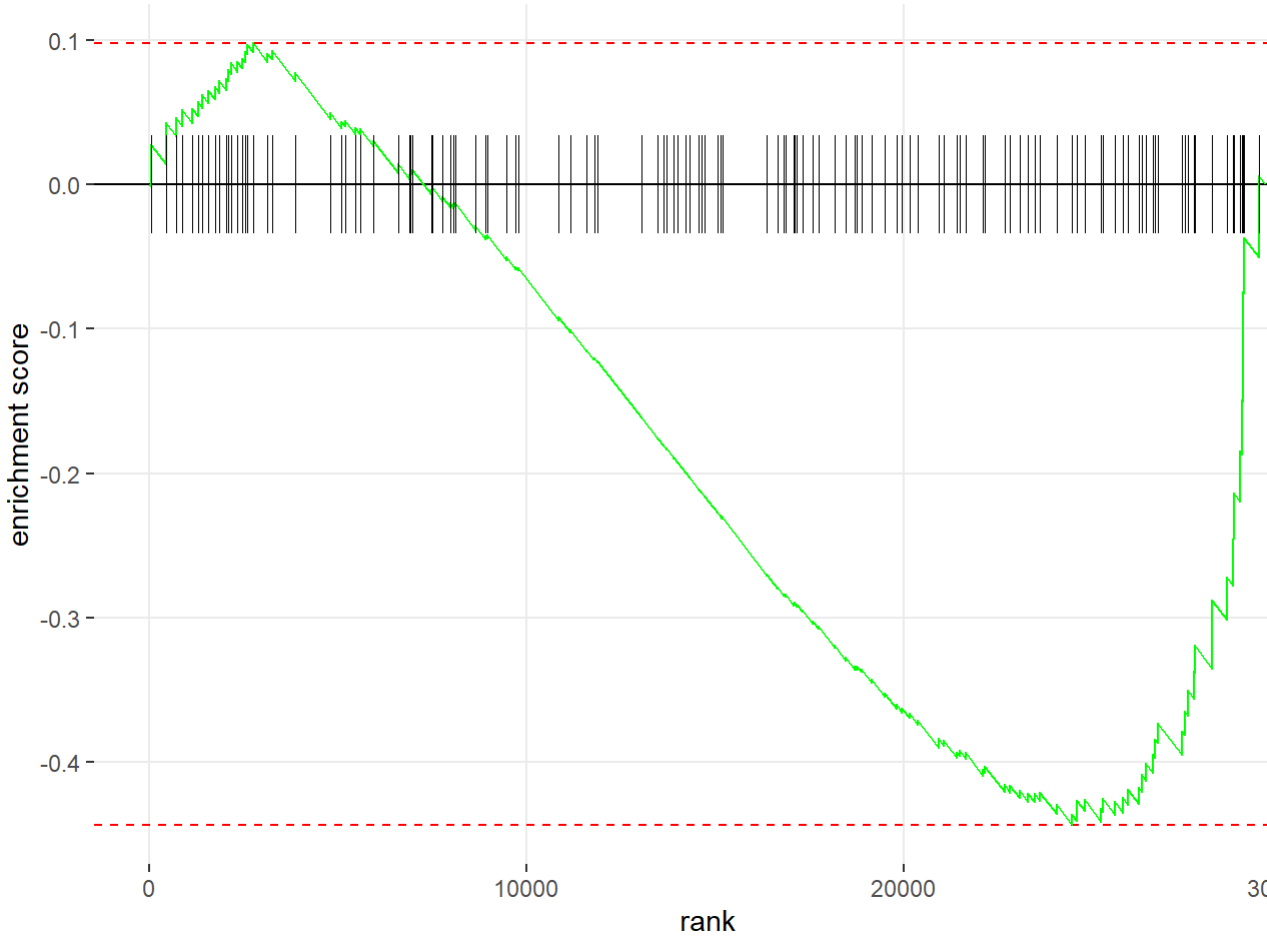
20: -1.1349580 266 32 20254,12614,23872,19206,20562,21887,...

21: -1.1009002 298 109 17308,20182,22042,11910,14228,19266,...
22: -1.1126936 278 132 20255,20266,20186,19279,20230,14009,...
23: -1.0878712 318 146 20193,12505,22035,21936,20274,26410,...
24: -1.0835194 332 146 20198,19017,14735,14734,12389,11910,...
25: -1.0577851 393 95 22042,12359,22173,18174,17859,13034,...
26: -1.0266799 463 157 20229,12505,14579,19267,11910,13636,...
27: -0.9552824 562 82 19687,26403,16800,20742,13589,15185,...
28: -0.9105065 576 42 17130,20742,12577,26409
29: -0.8809989 724 119 19016,20168,12359,11933,16775,26379,...
30: -0.8896358 712 110 20276,12505,12614,14228,17863,12308,...
31: -0.8670907 757 128 20276,12505,12389,20202,14228,17863,...
32: -0.9507260 569 63 20249,20229,18048,14679,16410,18095,...
33: -0.8647648 756 127 22035,12369,19246,19255,12362,20293,...
34: -0.8548462 708 72 22027,19247,11908,12566,17347,19087,...
35: -0.8880940 720 122 27966,12544,20174,27979,12566,18673,...
36: -1.0156091 490 150 20266,15975,22035,14734,19267,20288,...
37: -0.9922698 522 119 12035,12359,22173,11938,14450,20733,...
38: -0.9308811 637 103 20194,12613,19286,15488,17161,19270,...
39: -0.8944990 668 82 11949,11950,11946,11947,12369,11931,...
40: -0.9267902 639 107 14735,12505,14734,14228,17859,23996,...
41: -0.8834667 560 31 20198,20750,16410,16956,22325,12832
42: -0.9352937 669 172 12035,18646,16161,16423,19264,20292,...
43: -0.8412676 817 156 20249,20190,19245,21952,17883,17967,...
44: -0.7409806 662 21 20528,11630,14462,20521,21838
45: -0.7486367 682 27 12359,11927,16985,21672,22166,19122,...
46: 0.7356629 166 21 14534,14369,18131,18128,22418,22413,...
47: -0.7372620 892 110 19687,22042,20937,12419,20382,17350,...
48: -0.6609388 942 132 12544,20937,20382,12615,21887,14056,...
49: -0.6306158 846 40 16423,12362,16149,15945,17858,19124,...
50: -0.5229911 877 37 19206,18222,16842,19015
NES nMoreExtreme size leadingEdge

```
tmpInd <- match(pathways[["HALLMARK_APOPTOSIS"]],names(ranks))
tmpInd <- tmpInd[!is.na(tmpInd)]
ranks2 <- rep(0,length(ranks))
ranks2[tmpInd] <- ranks[tmpInd]
barplot(ranks2)
```




```
plotEnrichment(pathways[["HALLMARK_APOPTOSIS"]],
ranks)
```



GSEA article

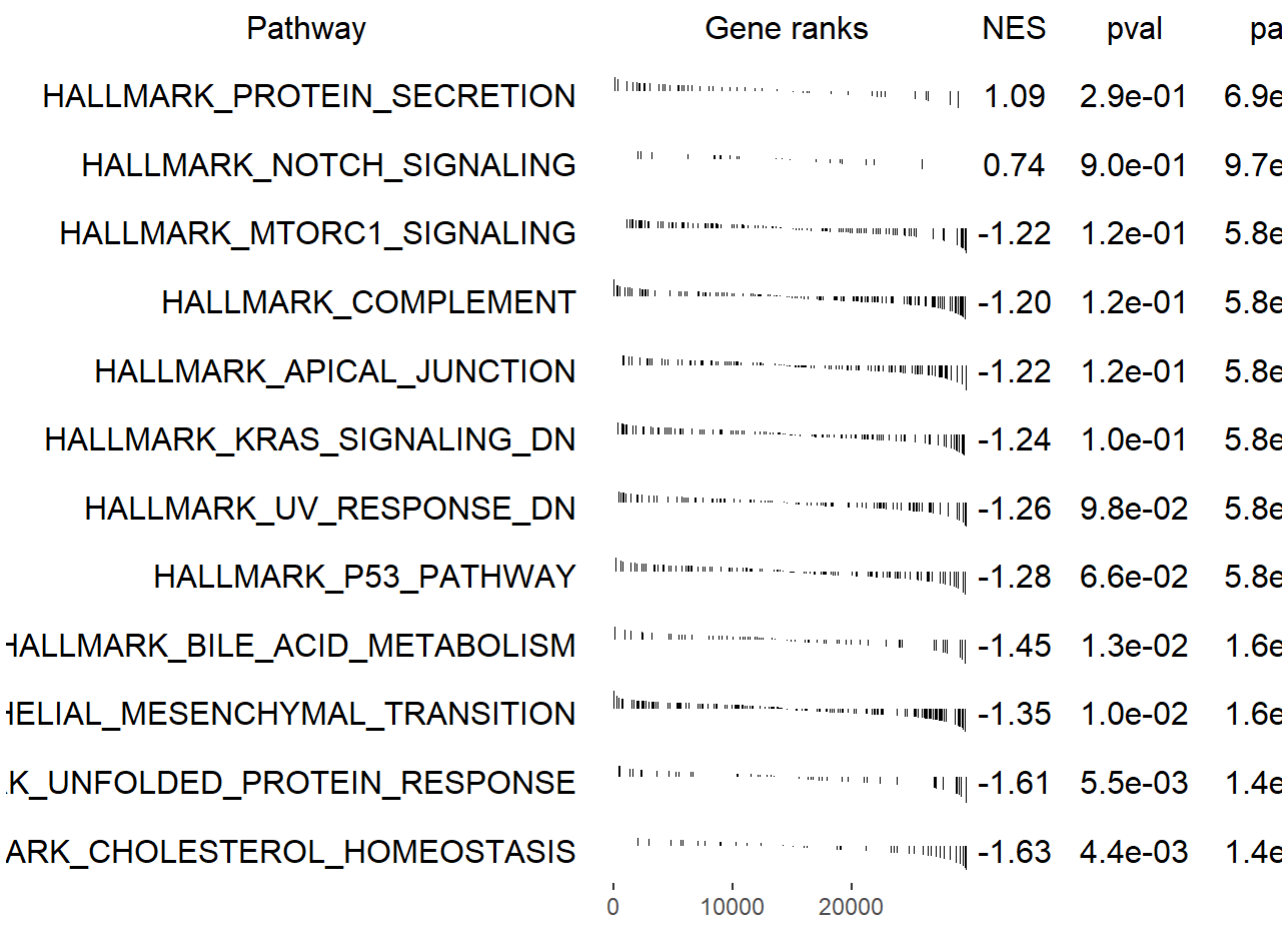
topPathwaysUp <- fgseaRes[ES > 0][head(order(pval), n=10), pathway]

topPathwaysDown <- fgseaRes[ES < 0][head(order(pval), n=10), pathway]

topPathways <- c(topPathwaysUp, rev(topPathwaysDown))

plotGseaTable(pathways[topPathways], ranks, fgseaRes,

gseaParam = 0.5)



7.3. CAMERA.

Altres bases de dades de conjunts de gens disponibles són procedents de la MSigDB del Broad Institute (Broad Institute's Molecular Signatures Database (MSigDB). CAMERA). CAMERA és una bona opció per contrastar un gran nombre de conjunts de gens, com els conjunts MSigDB, ja que és molt ràpid. Té l'avantatge de comptabilitzar la correlació intergènica de cada conjunt genètic (Wu i Smyth 2012) .

Aquí utilitzarem els conjunts de gens C2 per al ratolí, disponibles com a fitxers .rdata de la pàgina de bioinformàtica WEHI http://bioinf.wehi.edu.au/software/MSigDB/index.html . Els conjunts de gens C2 contenen 4725 conjunts genètics de diversos llocs: BioCarta, KEGG, Pathway Interaction Database, Reactome, així com alguns estudis publicats. No inclou termes GO.

load("mouse_c2_v5.rdata")

names(Mm.c2)[1:5]

[1] "KEGG_GLYCOLYSIS_GLUCONEOGENESIS"

[2] "KEGG_CITRATE_CYCLE_TCA_CYCLE"

[3] "KEGG_PENTOSE_PHOSPHATE_PATHWAY"

[4] "KEGG_PENTOSE_AND_GLUCURONATE_INTERCONVERSIONS"

[5] "KEGG_FRUCTOSE_AND_MANNOSE_METABOLISM"

length(Mm.c2)

[1] 4725

Els identificadors del gen són Entrez Gene ID. Hem d'assignar els ID del gen Entrez entre la llista de conjunts de gens i la nostra llista. Podem fer-ho mitjançant la funció ids2indices().

c2.ind <- ids2indices(Mm.c2, rownames(d2$counts))

De manera predeterminada, CAMERA pot estimar la correlació per a cada conjunt de gens per separat. Tanmateix, a la pràctica, funciona bé per establir una petita correlació intergència d´aprox. 0,05 mitjançant l' argument *inter.gene.cor*.

gst.camera<- camera.DGEList(d2,index=c2.ind,design=design.mat,contrast=2,inter.gene.cor=0.05)

Obtenim un dataframe de l´estadístic resultant, on cada fila correspòn a un grup de gens, ordenat pel p-value (encapçalat pel més significatiu). Mirem els 5 primers:

gst.camera[1:5,]

NGenes Direction PValue

REACTOME_ION_TRANSPORT_BY_P_TYPE_ATPASES 16 Up 0.001735217

MATZUK_POST-IMPLANTATION_AND_POST-PARTUM 10 Up 0.001813739

REACTOME_DIGESTION_OF_DIETARY_CARBOHYDRATE 1 Up 0.001978125

BIOCARTA_PLATELETAPP_PATHWAY 12 Down 0.003276654

REACTOME_ACTIVATED_POINT_MUTANTS_OF_FGFR2 13 Down 0.005884353

FDR

REACTOME_ION_TRANSPORT_BY_P_TYPE_ATPASES 0.9998271

MATZUK_POST-IMPLANTATION_AND_POST-PARTUM 0.9998271

REACTOME_DIGESTION_OF_DIETARY_CARBOHYDRATE 0.9998271

BIOCARTA_PLATELETAPP_PATHWAY 0.9998271

REACTOME_ACTIVATED_POINT_MUTANTS_OF_FGFR2 0.9998271

El nombre total de genes significatius al 5% de FDR és:

```
table(gst.camera$FDR < 0.05)
```

```
##
```

```
## FALSE
```

```
## 4718
```

```
write.csv(gst.camera,file="gst.csv")
```

Fem ara el contrast amb mouse_H_v5.rdata (hallmark datasets):

```
load("mouse_H_v5.rdata")
```

```
H.ind <- ids2indices(Mm.H, rownames(d2$counts))
```

```
H.camera <- camera.DGEList(d2,index=H.ind,design=design.mat,contrast=2,inter.gene.cor=0.05)
```

```
table(H.camera$FDR < 0.05)
```

```
##
```

```
## FALSE
```

```
## 50
```

```
H.camera[1:10,]
```

NGenes Direction PValue FDR

HALLMARK_NOTCH_SIGNALING 21 Down 0.1756169 0.9846482

HALLMARK_CHOLESTEROL_HOMEOSTASIS 39 Up 0.2520821 0.9846482

HALLMARK_OXIDATIVE_PHOSPHORYLATION 82 Up 0.4265903 0.9846482

HALLMARK_INTERFERON_ALPHA_RESPONSE 40 Down 0.4680038 0.9846482

HALLMARK_MTORC1_SIGNALING 120 Down 0.4726916 0.9846482

HALLMARK_FATTY_ACID_METABOLISM 103 Down 0.4847089 0.9846482

HALLMARK_PANCREAS_BETA_CELLS 32 Down 0.4923096 0.9846482

HALLMARK_HEDGEHOG_SIGNALING 32 Up 0.4937537 0.9846482

HALLMARK_UNFOLDED_PROTEIN_RESPONSE 44 Up 0.5392299 0.9846482

HALLMARK_IL6_JAK_STAT3_SIGNALING 75 Up 0.5593583 0.9846482

8. Proves de significació biològica autònomes.

8.1. ROAST.

ROAST és un exemple d'un test de gens autònom (Wu et al. 2010). Es fa la pregunta: "Els gens del meu conjunt solen expressar-se de manera diferent entre les meves condicions d'interès?". ROAST no utilitza informació sobre els altres gens de l'experiment, a diferència de CAMERA. ROAST és una bona opció per a quan estem interessats en un conjunt específic o en alguns conjunts. Realment no s´utilitza per contrastar milers de conjunts alhora (https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btq401).

H.camera[1:10,]

NGenes Direction PValue FDR

HALLMARK_NOTCH_SIGNALING 21 Down 0.1756169 0.9846482

HALLMARK_CHOLESTEROL_HOMEOSTASIS 39 Up 0.2520821 0.9846482

HALLMARK_OXIDATIVE_PHOSPHORYLATION 82 Up 0.4265903 0.9846482

HALLMARK_INTERFERON_ALPHA_RESPONSE 40 Down 0.4680038 0.9846482

HALLMARK_MTORC1_SIGNALING 120 Down 0.4726916 0.9846482

HALLMARK_FATTY_ACID_METABOLISM 103 Down 0.4847089 0.9846482

HALLMARK_PANCREAS_BETA_CELLS 32 Down 0.4923096 0.9846482

HALLMARK_HEDGEHOG_SIGNALING 32 Up 0.4937537 0.9846482

HALLMARK_UNFOLDED_PROTEIN_RESPONSE 44 Up 0.5392299 0.9846482

HALLMARK_IL6_JAK_STAT3_SIGNALING 75 Up 0.5593583 0.9846482

Com hem obtingut un gen implicat a l´apoptosi, anem a comprovar si hi ha més en MsigDB C2:

apop <- grep("APOPTOSIS_",names(c2.ind))

names(c2.ind)[apop]

[1] "BIOCARTA_TCAPOPTOSIS_PATHWAY"

[2] "REACTOME_APOPTOSIS_INDUCED_DNA_FRAGMENTATION"

[3] "HOLLMANN_APOPTOSIS_VIA_CD40_UP"

[4] "HOLLMANN_APOPTOSIS_VIA_CD40_DN"

[5] "LAU_APOPTOSIS_CDKN2A_UP"

[6] "LAU_APOPTOSIS_CDKN2A_DN"

[7] "GRAESSMANN_APOPTOSIS_BY_SERUM_DEPRIVATION_UP"

[8] "GRAESSMANN_APOPTOSIS_BY_SERUM_DEPRIVATION_DN"

[9] "CONCANNON_APOPTOSIS_BY_EPOXOMICIN_UP"

[10] "CONCANNON_APOPTOSIS_BY_EPOXOMICIN_DN"

[11] "GRAESSMANN_APOPTOSIS_BY_DOXORUBICIN_UP"

[12] "GRAESSMANN_APOPTOSIS_BY_DOXORUBICIN_DN"

[13] "HAMAI_APOPTOSIS_VIA_TRAIL_UP"

[14] "HAMAI_APOPTOSIS_VIA_TRAIL_DN"

[15] "RAMJAUN_APOPTOSIS_BY_TGFB1_VIA_SMAD4_UP"

[16] "RAMJAUN_APOPTOSIS_BY_TGFB1_VIA_SMAD4_DN"

[17] "RAMJAUN_APOPTOSIS_BY_TGFB1_VIA_MAPK1_UP"

[18] "RAMJAUN_APOPTOSIS_BY_TGFB1_VIA_MAPK1_DN"

[19] "DUTTA_APOPTOSIS_VIA_NFKB"

[20] "BROCKE_APOPTOSIS_REVERSED_BY_IL6"

[21] "DEBIASI_APOPTOSIS_BY_REOVIRUS_INFECTION_UP"

[22] "DEBIASI_APOPTOSIS_BY_REOVIRUS_INFECTION_DN"

[23] "WU_APOPTOSIS_BY_CDKN1A_VIA_TP53"

[24] "WU_APOPTOSIS_BY_CDKN1A_NOT_VIA_TP53"

Utilitzem ROAST per veure si aquests conjunts de gens relacionats amb "APOPTOSIS" solen expressar-se de manera diferent(la sintaxi DE `camera` I `roast` és pràcticament idèntica).

apop.rst <- roast.DGEList(d2,index=c2.ind[apop],design=design.mat,contrast=2,nrot=999)

apop.rst[1:15,]

NGenes PropDown PropUp Direction

GRAESSMANN_APOPTOSIS_BY_DOXORUBICIN_DN 635 1 0 Down

GRAESSMANN_APOPTOSIS_BY_DOXORUBICIN_UP 459 1 0 Down

GRAESSMANN_APOPTOSIS_BY_SERUM_DEPRIVATION_UP 252 1 0 Down

HAMAI_APOPTOSIS_VIA_TRAIL_UP 208 1 0 Down

DEBIASI_APOPTOSIS_BY_REOVIRUS_INFECTION_UP 143 1 0 Down

DEBIASI_APOPTOSIS_BY_REOVIRUS_INFECTION_DN 136 1 0 Down

CONCANNON_APOPTOSIS_BY_EPOXOMICIN_UP 133 1 0 Down

GRAESSMANN_APOPTOSIS_BY_SERUM_DEPRIVATION_DN 103 1 0 Down

HOLLMANN_APOPTOSIS_VIA_CD40_DN 99 1 0 Down

HOLLMANN_APOPTOSIS_VIA_CD40_UP 95 1 0 Down

CONCANNON_APOPTOSIS_BY_EPOXOMICIN_DN 87 1 0 Down

BROCKE_APOPTOSIS_REVERSED_BY_IL6 83 1 0 Down

HAMAI_APOPTOSIS_VIA_TRAIL_DN 75 1 0 Down

LAU_APOPTOSIS_CDKN2A_UP 38 1 0 Down

WU_APOPTOSIS_BY_CDKN1A_VIA_TP53 31 1 0 Down

PValue FDR PValue.Mixed

GRAESSMANN_APOPTOSIS_BY_DOXORUBICIN_DN 0.001 0.001 0.001

GRAESSMANN_APOPTOSIS_BY_DOXORUBICIN_UP 0.001 0.001 0.001

GRAESSMANN_APOPTOSIS_BY_SERUM_DEPRIVATION_UP 0.001 0.001 0.001

HAMAI_APOPTOSIS_VIA_TRAIL_UP 0.001 0.001 0.001

DEBIASI_APOPTOSIS_BY_REOVIRUS_INFECTION_UP 0.001 0.001 0.001

DEBIASI_APOPTOSIS_BY_REOVIRUS_INFECTION_DN 0.001 0.001 0.001

CONCANNON_APOPTOSIS_BY_EPOXOMICIN_UP 0.001 0.001 0.001

GRAESSMANN_APOPTOSIS_BY_SERUM_DEPRIVATION_DN 0.001 0.001 0.001

HOLLMANN_APOPTOSIS_VIA_CD40_DN 0.001 0.001 0.001

HOLLMANN_APOPTOSIS_VIA_CD40_UP 0.001 0.001 0.001

CONCANNON_APOPTOSIS_BY_EPOXOMICIN_DN 0.001 0.001 0.001

BROCKE_APOPTOSIS_REVERSED_BY_IL6 0.001 0.001 0.001

HAMAI_APOPTOSIS_VIA_TRAIL_DN 0.001 0.001 0.001

LAU_APOPTOSIS_CDKN2A_UP 0.001 0.001 0.001

WU_APOPTOSIS_BY_CDKN1A_VIA_TP53 0.001 0.001 0.001

FDR.Mixed

GRAESSMANN_APOPTOSIS_BY_DOXORUBICIN_DN 0.001

GRAESSMANN_APOPTOSIS_BY_DOXORUBICIN_UP 0.001

GRAESSMANN_APOPTOSIS_BY_SERUM_DEPRIVATION_UP 0.001

HAMAI_APOPTOSIS_VIA_TRAIL_UP 0.001

DEBIASI_APOPTOSIS_BY_REOVIRUS_INFECTION_UP 0.001

DEBIASI_APOPTOSIS_BY_REOVIRUS_INFECTION_DN 0.001

CONCANNON_APOPTOSIS_BY_EPOXOMICIN_UP 0.001

GRAESSMANN_APOPTOSIS_BY_SERUM_DEPRIVATION_DN 0.001

HOLLMANN_APOPTOSIS_VIA_CD40_DN 0.001

HOLLMANN_APOPTOSIS_VIA_CD40_UP 0.001

CONCANNON_APOPTOSIS_BY_EPOXOMICIN_DN 0.001

BROCKE_APOPTOSIS_REVERSED_BY_IL6 0.001

HAMAI_APOPTOSIS_VIA_TRAIL_DN 0.001

LAU_APOPTOSIS_CDKN2A_UP 0.001

WU_APOPTOSIS_BY_CDKN1A_VIA_TP53 0.001

La columna NGenes indica el nombre de gens de cada conjunt. Les columnes PropDown i PropUp contenen les proporcions de gens del conjunt que estan regulades a la baixa i a l´alça, respectivament, amb canvis de plecs absoluts superiors a 2. La direcció neta del canvi es determina a partir de la significació dels canvis en cada direcció i es mostra a la columna Direcció. El PValue proporciona evidències de si la majoria de gens del conjunt són expressats diferencialment en la direcció especificada PValue.Mixed proporciona evidències de si la majoria de gens del conjunt són expressats diferencialment en qualsevol direcció. Les FDR es calculen a partir dels valors p corresponents a tots els conjunts.