Anàlisi de dades en experiments de RNAseq

Estadística per a les Biociències

2 of març, 2023

Índex

```
if(!require(BiocManager)){
      install.packages("BiocManager", dep=TRUE)
}
```

```
## Loading required package: BiocManager
```

```
## Bioconductor version '3.15' is out-of-date; the current release version '3.16'
##   is available with R version '4.2'; see https://bioconductor.org/install
```

```
installifnot <- function (pckgName, BioC=TRUE){
  if(BioC){
    if(!require(pckgName, character.only=TRUE)){
      BiocManager::install(pckgName)
    }
  }else{
    if(!require(pckgName, character.only=TRUE)){
      install.packages(pckgName, dep=TRUE)
    }
  }
}

libraries<- c("limma","edgeR","Glimma","Mus.musculus")
for(i in libraries){
  print(i)
  installifnot(i)
  library(i, character.only = TRUE, quietly = TRUE)
}
```

```
## [1] "limma"
```

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

```
## [1] "edgeR"
```

```
## Loading required package: edgeR
```

```
## [1] "Glimma"
```

```
## Loading required package: Glimma
```

```
## Bioconductor version 3.15 (BiocManager 1.30.18), R 4.2.1 (2022-06-23 ucrt)
```

```
## Installing package(s) 'Glimma'
```

```
## also installing the dependencies 'plogr', 'lambda.r', 'futile.options', 'RSQLite', 'KEGGREST', 'XML'
```

```
## package 'plogr' successfully unpacked and MD5 sums checked
## package 'lambda.r' successfully unpacked and MD5 sums checked
## package 'futile.options' successfully unpacked and MD5 sums checked
## package 'RSQLite' successfully unpacked and MD5 sums checked
## package 'KEGGREST' successfully unpacked and MD5 sums checked
## package 'XML' successfully unpacked and MD5 sums checked
## package 'futile.logger' successfully unpacked and MD5 sums checked
## package 'snow' successfully unpacked and MD5 sums checked
## package 'AnnotationDbi' successfully unpacked and MD5 sums checked
## package 'annotate' successfully unpacked and MD5 sums checked
## package 'BiocParallel' successfully unpacked and MD5 sums checked
## package 'genefilter' successfully unpacked and MD5 sums checked
## package 'geneplotter' successfully unpacked and MD5 sums checked
## package 'DESeq2' successfully unpacked and MD5 sums checked
```

```
## package 'Glimma' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Esteban Vegas\AppData\Local\Temp\RtmpMZ3O14\downloaded_packages
```

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

```
## Old packages: 'ade4', 'BiocManager', 'bookdown', 'broom', 'bslib', 'C50',
##   'cachem', 'chron', 'classInt', 'cli', 'collapse', 'colorspace', 'Cubist',
##   'curl', 'DALEX', 'data.table', 'dbplyr', 'DescTools', 'digest', 'dplyr',
##   'DT', 'dtplyr', 'e1071', 'emmeans', 'evaluate', 'fansi', 'fastmap',
##   'flexdashboard', 'flextable', 'fontawesome', 'forcats', 'forecast',
##   'formatR', 'Formula', 'fs', 'future', 'gargle', 'gdtools', 'ggiraph',
##   'ggplot2', 'ggpubr', 'ggrepel', 'gh', 'gimme', 'gower', 'haven', 'highr',
##   'htmltools', 'htmlwidgets', 'httpuv', 'httr', 'igraph', 'ingredients',
##   'insight', 'isoband', 'keras', 'kernlab', 'knitr', 'lava', 'lavaan',
##   'listenv', 'locfit', 'lubridate', 'markdown', 'MASS', 'naniar', 'nlme',
##   'officer', 'OpenImageR', 'packrat', 'parallelly', 'partykit', 'pbapply',
##   'pbkrtest', 'progressr', 'pryr', 'purrr', 'questionr', 'ragg', 'Rcpp',
##   'RcppArmadillo', 'RcppTOML', 'RCurl', 'readODS', 'readr', 'readxl',
##   'recipes', 'reticulate', 'rgl', 'RhpcBLASctl', 'rmarkdown', 'rsconnect',
##   'rstatix', 'rstpm2', 'sass', 'shiny', 'sourcetools', 'sp', 'stringi',
##   'styler', 'survival', 'svglite', 'tensorflow', 'tidyr', 'tidyverse',
##   'timechange', 'timeDate', 'tinytex', 'tokenizers', 'utf8', 'vcd', 'vctrs',
##   'vegan3d', 'visdat', 'vroom', 'writexl', 'xfun', 'yaml'
```

```
## [1] "Mus.musculus"
```

```
## Loading required package: Mus.musculus
```

```
## Bioconductor version 3.15 (BiocManager 1.30.18), R 4.2.1 (2022-06-23 ucrt)
```

```
## Installing package(s) 'Mus.musculus'
```

```
## also installing the dependencies 'Rhtslib', 'rjson', 'filelock', 'Rsamtools', 'GenomicAlignments', '
```

```
## package 'Rhtslib' successfully unpacked and MD5 sums checked
## package 'rjson' successfully unpacked and MD5 sums checked
## package 'filelock' successfully unpacked and MD5 sums checked
## package 'Rsamtools' successfully unpacked and MD5 sums checked
## package 'GenomicAlignments' successfully unpacked and MD5 sums checked
## package 'restfulr' successfully unpacked and MD5 sums checked
## package 'BiocFileCache' successfully unpacked and MD5 sums checked
## package 'graph' successfully unpacked and MD5 sums checked
## package 'RBGL' successfully unpacked and MD5 sums checked
## package 'BiocIO' successfully unpacked and MD5 sums checked
## package 'rtracklayer' successfully unpacked and MD5 sums checked
## package 'biomaRt' successfully unpacked and MD5 sums checked
## package 'OrganismDbi' successfully unpacked and MD5 sums checked
## package 'GenomicFeatures' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Esteban Vegas\AppData\Local\Temp\RtmpMZ3O14\downloaded_packages
```

```
## installing the source packages 'GO.db', 'org.Mm.eg.db', 'TxDb.Mmusculus.UCSC.mm10.knownGene', 'Mus.mu

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

## Old packages: 'ade4', 'BiocManager', 'bookdown', 'broom', 'bslib', 'C50',
##   'cachem', 'chron', 'classInt', 'cli', 'collapse', 'colorspace', 'Cubist',
##   'curl', 'DALEX', 'data.table', 'dbplyr', 'DescTools', 'digest', 'dplyr',
##   'DT', 'dtplyr', 'e1071', 'emmeans', 'evaluate', 'fansi', 'fastmap',
##   'flexdashboard', 'flextable', 'fontawesome', 'forcats', 'forecast',
##   'formatR', 'Formula', 'fs', 'future', 'gargle', 'gdtools', 'ggiraph',
##   'ggplot2', 'ggpubr', 'ggrepel', 'gh', 'gimme', 'gower', 'haven', 'highr',
##   'htmltools', 'htmlwidgets', 'httpuv', 'httr', 'igraph', 'ingredients',
##   'insight', 'isoband', 'keras', 'kernlab', 'knitr', 'lava', 'lavaan',
##   'listenv', 'locfit', 'lubridate', 'markdown', 'MASS', 'naniar', 'nlme',
##   'officer', 'OpenImageR', 'packrat', 'parallelly', 'partykit', 'pbapply',
##   'pbkrtest', 'progressr', 'pryr', 'purrr', 'questionr', 'ragg', 'Rcpp',
##   'RcppArmadillo', 'RcppTOML', 'RCurl', 'readODS', 'readr', 'readxl',
##   'recipes', 'reticulate', 'rgl', 'RhpcBLASctl', 'rmarkdown', 'rsconnect',
##   'rstatix', 'rstpm2', 'sass', 'shiny', 'sourcetools', 'sp', 'stringi',
##   'styler', 'survival', 'svglite', 'tensorflow', 'tidyr', 'tidyverse',
##   'timechange', 'timeDate', 'tinytex', 'tokenizers', 'utf8', 'vcd', 'vctrs',
##   'vegan3d', 'visdat', 'vroom', 'writexl', 'xfun', 'yaml'

##
## Attaching package: 'BiocGenerics'

## The following object is masked from 'package:limma':
##
##     plotMA

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
```

```
##     expand.grid, I, unname
```

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

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

```
##
```

```
##
```

```
libraries <- c("RColorBrewer","gplots")

for(i in libraries){
  installifnot(i, BioC = FALSE)
  library(i, character.only = TRUE, quietly = TRUE)
}
```

```
## Loading required package: RColorBrewer
```

```
## Loading required package: gplots
```

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

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

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

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

Introducció

L'experiment que s'analitza en aquest treball prové de Sheridan et al. (2015) i consta de tres poblacions cel · lulars: basals, progenitor luminal (LP) i luminal madur (ML), obtinguts a partir de les glàndules mamàries de ratolins femella, cadascun replicat per triplicat. Les mostres d'ARN es van seqüenciar en tres batch en un Illumina HiSeq 2000 per obtenir-ne `100 base-pair single-end reads`.

L'anàlisi en aquesta pràctica suposa que els `reads` obtinguts en l'experiment RNA-seq s'han alineat a un genoma de referència adequat i s'han quantificat en `counts` associats a gens.

En aquest cas, els `reads` es van alinear amb el genoma de referència del ratolí (mm10) mitjançant el pipeline basat en R, disponible al paquet `Rsubread`, específicament la funció `align` seguida de la funció `featureCounts` per obtenir els `counts`.

Preparar les dades

Per començar amb aquesta anàlisi, descarrega el fitxer GSE63310_RAW.tar des de

http://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE63310&format=file,

i extraieu-ne els fitxers. Opcionalment, pots trobar els fitxers en la carpeta de treball en el campus atenea.

Cadascun d'aquests arxius `txt` contenen els counts en brut (raw) a nivell de gen de cada mostra (ratolí). De manera descriptiva, pots observar les 20 primeres files del primer fitxer.

Importar les dades

```
#path <- "dadesRNAseq1"
files <- c("GSM1545535_10_6_5_11.txt", "GSM1545536_9_6_5_11.txt",
           "GSM1545538_purep53.txt", "GSM1545539_JMS8-2.txt",
           "GSM1545540_JMS8-3.txt", "GSM1545541_JMS8-4.txt",
           "GSM1545542_JMS8-5.txt", "GSM1545544_JMS9-P7c.txt",
           "GSM1545545_JMS9-P8c.txt")

read.delim(paste(params$path,files[1],sep="/"), nrow = 20)
```

```
##     EntrezID GeneLength Count
## 1     497097       3634     1
## 2  100503874       3259     0
## 3  100038431       1634     0
## 4      19888       9747     0
## 5      20671       3130     1
## 6      27395       4203   431
## 7      18777       2433   768
## 8  100503730        799     4
## 9      21399       2847   810
## 10     58175       2241   452
## 11    108664       1976  1716
## 12     18387       4707     0
## 13    226304       3692     0
## 14     12421       7046  3451
## 15    620393        858     0
## 16    240690       6161     0
## 17    319263       5232  2026
## 18     71096       4193     0
## 19     59014       2048   956
## 20     76187       3139    54
```

Si bé, cadascun dels nou fitxers `txt` es pot llegir en R per separat i combinar-los en una matriu d'expressió amb els counts, el paquet `edgeR` ofereix una forma convenient de fer-ho en un sol pas mitjançant la funció `readDGE`.

```
x <- readDGE(paste(params$path,files,sep="/"), columns = c(1, 3))
```

L'objecte `DGEList` que en resulta conté una matriu de counts amb 27179 files associades amb identificadors dels gens en notació Entrez (EntrezID) i 9 columnes associades amb les mostres de l'experiment.

```
class(x)
```

```
## [1] "DGEList"
## attr(,"package")
## [1] "edgeR"
```

```
dim(x)
```

```
## [1] 27179     9
```

```
names(x)
```

```
## [1] "samples" "counts"
```

```
str(x)
```

```
## Formal class 'DGEList' [package "edgeR"] with 1 slot
##   ..@ .Data:List of 2
##   .. ..$ :'data.frame':  9 obs. of  4 variables:
##   .. .. ..$ files       : chr [1:9] "dadesRNAseq1/GSM1545535_10_6_5_11.txt" "dadesRNAseq1/GSM1545536
##   .. .. ..$ group       : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1
##   .. .. ..$ lib.size    : num [1:9] 32863052 35335491 57160817 51368625 75795034 ...
##   .. .. ..$ norm.factors: num [1:9] 1 1 1 1 1 1 1 1 1
##   .. ..$ : num [1:27179, 1:9] 1 0 0 0 1 431 768 4 810 452 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ Tags   : chr [1:27179] "497097" "100503874" "100038431" "19888" ...
##   .. .. .. ..$ Samples: chr [1:9] "dadesRNAseq1/GSM1545535_10_6_5_11" "dadesRNAseq1/GSM1545536_9_6_5
##   ..$ names: chr [1:2] "samples" "counts"
```

Anotar les mostres

Per seguir amb l'anàlisi, cal associar la informació a nivell de les mostres relacionada amb el disseny experimental amb les columnes de la matriu de counts. Això ha d'incloure variables experimentals, tant biològiques com tècniques que poden tenir un efecte sobre els nivells d'expressió. Per exemple, són el tipus de cel.lula (basal, LP i ML en aquest experiment), el fenotip (estat de la malaltia, sexe, edat), tractament de mostres (fàrmac, control) i informació per lots (la data en que es va realitzar l'experiment si es van recollir mostres i analitzar-les en diferents moments) per citar només algunes.

El nostre objecte `DGEList` conté un `data.frame` amb les dades de les mostres que emmagatzema tant el tipus de cel.lular (`group`) com el batch (`lane`), cadascun dels quals consta de tres nivells. Tingueu en compte que dins de `x$samples` la mida de la biblioteca (`lib.size`) es calcula automàticament per a cada mostra i els factors de normalització s'estableixen en 1. Per senzillesa, eliminem el GEO ID (`GSM *`) dels noms de columnes del nostre objecte `DGEList x`.

```
x$samples
```

```
##                                                                files group
## dadesRNAseq1/GSM1545535_10_6_5_11 dadesRNAseq1/GSM1545535_10_6_5_11.txt     1
## dadesRNAseq1/GSM1545536_9_6_5_11   dadesRNAseq1/GSM1545536_9_6_5_11.txt     1
## dadesRNAseq1/GSM1545538_purep53     dadesRNAseq1/GSM1545538_purep53.txt     1
## dadesRNAseq1/GSM1545539_JMS8-2       dadesRNAseq1/GSM1545539_JMS8-2.txt     1
## dadesRNAseq1/GSM1545540_JMS8-3       dadesRNAseq1/GSM1545540_JMS8-3.txt     1
## dadesRNAseq1/GSM1545541_JMS8-4       dadesRNAseq1/GSM1545541_JMS8-4.txt     1
## dadesRNAseq1/GSM1545542_JMS8-5       dadesRNAseq1/GSM1545542_JMS8-5.txt     1
## dadesRNAseq1/GSM1545544_JMS9-P7c   dadesRNAseq1/GSM1545544_JMS9-P7c.txt     1
## dadesRNAseq1/GSM1545545_JMS9-P8c   dadesRNAseq1/GSM1545545_JMS9-P8c.txt     1
##                                  lib.size norm.factors
## dadesRNAseq1/GSM1545535_10_6_5_11 32863052            1
## dadesRNAseq1/GSM1545536_9_6_5_11  35335491            1
## dadesRNAseq1/GSM1545538_purep53   57160817            1
## dadesRNAseq1/GSM1545539_JMS8-2    51368625            1
## dadesRNAseq1/GSM1545540_JMS8-3    75795034            1
## dadesRNAseq1/GSM1545541_JMS8-4    60517657            1
## dadesRNAseq1/GSM1545542_JMS8-5    55086324            1
## dadesRNAseq1/GSM1545544_JMS9-P7c  21311068            1
## dadesRNAseq1/GSM1545545_JMS9-P8c  19958838            1
```

```
samplenames <- substring(colnames(x), 12, nchar(colnames(x)))
samplenames
```

```
## [1] "1/GSM1545535_10_6_5_11" "1/GSM1545536_9_6_5_11"  "1/GSM1545538_purep53"
## [4] "1/GSM1545539_JMS8-2"    "1/GSM1545540_JMS8-3"    "1/GSM1545541_JMS8-4"
## [7] "1/GSM1545542_JMS8-5"    "1/GSM1545544_JMS9-P7c"  "1/GSM1545545_JMS9-P8c"
```

```
colnames(x) <- samplenames
group <- as.factor(c("LP", "ML", "Basal", "Basal", "ML", "LP",
                     "Basal", "ML", "LP"))
x$samples$group <- group
lane <- as.factor(rep(c("L004", "L006", "L008"), c(3, 4, 2)))
x$samples$lane <- lane
x$samples
```

```
##                                                          files group lib.size
## 1/GSM1545535_10_6_5_11 dadesRNAseq1/GSM1545535_10_6_5_11.txt    LP 32863052
## 1/GSM1545536_9_6_5_11   dadesRNAseq1/GSM1545536_9_6_5_11.txt    ML 35335491
## 1/GSM1545538_purep53     dadesRNAseq1/GSM1545538_purep53.txt Basal 57160817
## 1/GSM1545539_JMS8-2       dadesRNAseq1/GSM1545539_JMS8-2.txt Basal 51368625
## 1/GSM1545540_JMS8-3       dadesRNAseq1/GSM1545540_JMS8-3.txt    ML 75795034
## 1/GSM1545541_JMS8-4       dadesRNAseq1/GSM1545541_JMS8-4.txt    LP 60517657
## 1/GSM1545542_JMS8-5       dadesRNAseq1/GSM1545542_JMS8-5.txt Basal 55086324
## 1/GSM1545544_JMS9-P7c   dadesRNAseq1/GSM1545544_JMS9-P7c.txt    ML 21311068
## 1/GSM1545545_JMS9-P8c   dadesRNAseq1/GSM1545545_JMS9-P8c.txt    LP 19958838
##                        norm.factors lane
## 1/GSM1545535_10_6_5_11            1 L004
## 1/GSM1545536_9_6_5_11             1 L004
## 1/GSM1545538_purep53              1 L004
## 1/GSM1545539_JMS8-2               1 L006
## 1/GSM1545540_JMS8-3               1 L006
## 1/GSM1545541_JMS8-4               1 L006
## 1/GSM1545542_JMS8-5               1 L006
## 1/GSM1545544_JMS9-P7c             1 L008
## 1/GSM1545545_JMS9-P8c             1 L008
```

Anotar els gens

Un segon `data.frame` anomenat `genes` en l'objecte `DGEList` s'utilitza per emmagatzemar informació dels gens associada a les files de la matriu d'expressió. Aquesta informació es pot obtenir utilitzant paquets específics de l'organisme com `Mus.musculus` per a ratolí (o `Homo.sapiens` per a humans) o el paquet `biomaRt` que accedeix a les bases de dades del genoma d'Ensembl per tal de realitzar una anotació genòmica. El tipus d'informació que es pot obtenir inclou símbols gènics, noms de gens, noms i posicions en els cromosomes, identificadors Entrez gene ID , Refseq gene ID i Ensembl gene ID per citar només alguns.

`biomaRt` principalment funciona amb els ID del gen d'Ensembl, mentre que `Mus.musculus` envia informació de diverses fonts i permet als usuaris escollir entre molts identificadors de gen diferents com a clau. Els ID de gen d'Entrez disponibles al nostre conjunt de dades es van anotar mitjançant el paquet `Mus.musculus` d'aquesta manera es va recuperar els identificadors dels gens i la informació cromosòmica.

```
#if (!requireNamespace("BiocManager", quietly = TRUE))
#  install.packages("BiocManager")
#BiocManager::install("Mus.musculus")
library("Mus.musculus") # Gene annotations for the Mus musculus genome
```

```
head(x$counts)
```

```
##            Samples
## Tags        1/GSM1545535_10_6_5_11 1/GSM1545536_9_6_5_11 1/GSM1545538_purep53
```

```
##   497097                          1                   2                 342
##   100503874                       0                   0                   5
##   100038431                       0                   0                   0
##   19888                           0                   1                   0
##   20671                           1                   1                  76
##   27395                         431                 771                1368
##            Samples
## Tags        1/GSM1545539_JMS8-2 1/GSM1545540_JMS8-3 1/GSM1545541_JMS8-4
##   497097                    526                   3                   3
##   100503874                   6                   0                   0
##   100038431                   0                   0                   0
##   19888                       0                  17                   2
##   20671                      40                  33                  14
##   27395                    1268                1564                 769
##            Samples
## Tags        1/GSM1545542_JMS8-5 1/GSM1545544_JMS9-P7c 1/GSM1545545_JMS9-P8c
##   497097                    535                   2                   0
##   100503874                   5                   0                   0
##   100038431                   1                   0                   0
##   19888                       0                   1                   0
##   20671                      98                  18                   8
##   27395                     818                 468                 342
```

```
dim(x$counts)
```

```
## [1] 27179     9
```

```
geneid <- rownames(x)
genes <- select(Mus.musculus, keys = geneid,
                columns = c("SYMBOL", "TXCHROM"),
                keytype = "ENTREZID")
```

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

```
head(genes)
```

```
##    ENTREZID  SYMBOL TXCHROM
## 1    497097    Xkr4    chr1
## 2 100503874 Gm19938    <NA>
## 3 100038431 Gm10568    <NA>
## 4     19888     Rp1    chr1
## 5     20671   Sox17    chr1
## 6     27395  Mrpl15    chr1
```

Com passa amb qualsevol anotació genòmica , els ID de gen d'Entrez poden no mapar de manera 1 a 1 amb la informació d'interès del gen. És important comprovar si hi ha identificadors de gen duplicats, comprendre la font de la duplicació i aclarir-la. La nostra anotació conté 28 gens que es corresponen amb múltiples cromosomes; per exemple, el microRNA Mir5098 estan associats a "chr5", "chr8", "chr11" i "chr17".

Resoldre els identificadors duplicats, es una tasca complexa que requeriria revisar el procés d'alineament en el genome de referència. Per senzillesa, en la pràctica d'avui seleccionarem un dels cromosomes per representar el gen amb una anotació duplicada, mantenint només la primera ocurrència de cadascuna identificació del gen.

```
sel<-which(duplicated(genes$ENTREZID))
genes[sel,]
```

```
##        ENTREZID        SYMBOL             TXCHROM
```

```
## 2774     268373          Ppia                chr11
## 5362  100316809     Mir1906-1                 chrX
## 7516     545056        Gm5801                chr14
## 9381     170658     Ndufs5-ps                chr16
## 9567      12228          Btg3                chr17
## 11546 100217457      Snord58b                chr18
## 11758     14109           Fau                chr19
## 11981    433224        Gm5512                chr19
## 16095 100042555       Gm13305 chr4_JH584293_random
## 16096 100042555       Gm13305 chr4_JH584294_random
## 16100 100042493        Ccl21b chr4_JH584293_random
## 16102    621580       Gm21953 chr4_GL456350_random
## 16110    545611      Fam205a2 chr4_GL456350_random
## 16115 100040048        Ccl27b chr4_JH584293_random
## 16121     24047         Ccl19 chr4_JH584294_random
## 17479 100861978          <NA> chr4_JH584293_random
## 17480 100861978          <NA> chr4_JH584294_random
## 18179    331195      Pramel39 chr5_GL456354_random
## 18180    331195      Pramel39 chr5_JH584298_random
## 18186 100041102      Pramel42 chr5_JH584296_random
## 18187 100041102      Pramel42 chr5_JH584297_random
## 18194    622894        Gm6367 chr5_JH584299_random
## 18203 100041354        Gm3286 chr5_GL456354_random
## 18208    666203      Pramel50 chr5_GL456354_random
## 19042    545762       Gm16367 chr5_JH584298_random
## 19043    545762       Gm16367 chr5_JH584299_random
## 22457    434233       Ppp1ccb                chr7
## 26601     30058       Timm8a1                chrX
## 26889    654820 G530011O06Rik                chrY
```

```
genes <- genes[!duplicated(genes$ENTREZID), ]
```

IMPORTANT. En aquest exemple, l'ordre dels gens és el mateix tant en l'anotació com en l'objecte de dades. Si aquest no és el cas degut faltants i/o identificadors de gen reorganitzats, la funció `match` es pot utilitzar per ordenar els gens correctament.

El dataframe amb les anotacions dels gens després s'afegeix a l'objecte de dades i s'empaqueta en un objecte `DGEList` que conté:

- quantificació d'expressió (raw counts),
- informació de les mostres,
- anotació dels gens.

```
x$genes <- genes
x
```

```
## An object of class "DGEList"
## $samples
##                                                        files group lib.size
## 1/GSM1545535_10_6_5_11 dadesRNAseq1/GSM1545535_10_6_5_11.txt    LP 32863052
## 1/GSM1545536_9_6_5_11   dadesRNAseq1/GSM1545536_9_6_5_11.txt    ML 35335491
## 1/GSM1545538_purep53     dadesRNAseq1/GSM1545538_purep53.txt Basal 57160817
## 1/GSM1545539_JMS8-2       dadesRNAseq1/GSM1545539_JMS8-2.txt Basal 51368625
## 1/GSM1545540_JMS8-3       dadesRNAseq1/GSM1545540_JMS8-3.txt    ML 75795034
## 1/GSM1545541_JMS8-4       dadesRNAseq1/GSM1545541_JMS8-4.txt    LP 60517657
## 1/GSM1545542_JMS8-5       dadesRNAseq1/GSM1545542_JMS8-5.txt Basal 55086324
## 1/GSM1545544_JMS9-P7c   dadesRNAseq1/GSM1545544_JMS9-P7c.txt    ML 21311068
```

```
## 1/GSM1545545_JMS9-P8c   dadesRNAseq1/GSM1545545_JMS9-P8c.txt    LP 19958838
##                        norm.factors lane
## 1/GSM1545535_10_6_5_11            1 L004
## 1/GSM1545536_9_6_5_11             1 L004
## 1/GSM1545538_purep53              1 L004
## 1/GSM1545539_JMS8-2               1 L006
## 1/GSM1545540_JMS8-3               1 L006
## 1/GSM1545541_JMS8-4               1 L006
## 1/GSM1545542_JMS8-5               1 L006
## 1/GSM1545544_JMS9-P7c             1 L008
## 1/GSM1545545_JMS9-P8c             1 L008
##
## $counts
##           Samples
## Tags        1/GSM1545535_10_6_5_11 1/GSM1545536_9_6_5_11 1/GSM1545538_purep53
##   497097                         1                     2                  342
##   100503874                      0                     0                    5
##   100038431                      0                     0                    0
##   19888                          0                     1                    0
##   20671                          1                     1                   76
##           Samples
## Tags        1/GSM1545539_JMS8-2 1/GSM1545540_JMS8-3 1/GSM1545541_JMS8-4
##   497097                    526                   3                   3
##   100503874                   6                   0                   0
##   100038431                   0                   0                   0
##   19888                       0                  17                   2
##   20671                      40                  33                  14
##           Samples
## Tags        1/GSM1545542_JMS8-5 1/GSM1545544_JMS9-P7c 1/GSM1545545_JMS9-P8c
##   497097                    535                   2                   0
##   100503874                   5                   0                   0
##   100038431                   1                   0                   0
##   19888                       0                   1                   0
##   20671                      98                  18                   8
## 27174 more rows ...
##
## $genes
##    ENTREZID  SYMBOL TXCHROM
## 1    497097    Xkr4    chr1
## 2 100503874 Gm19938    <NA>
## 3 100038431 Gm10568    <NA>
## 4     19888     Rp1    chr1
## 5     20671   Sox17    chr1
## 27174 more rows ...
```

Pre-processat de les dades

Transformació de les dades en cru

En l'anàlisi de l'expressió diferencial i aspectes relacionats, l'expressió gènica poques vegades es considera a nivell de `counts` en brut, atés que les biblioteques seqüenciades a major profunditat produiran major número de `counts`. Més aviat, És una pràctica habitual transformar els `counts` en brut a una escala relativa a la mida de les biblioteques. Les transformacions populars inclouen `counts` per milió (CPM), log2 `counts` per milió (log-CPM), `counts` per kilobase de transcripció per milió (RPKM), fragments per kilobase de transcripció

per milió (FPKM) i Transcripts per milió (TPM).

En l'anàlisis que plantejem en aquesta pràctica, utilitzarem les transformacions CPM i log-CPM que no tenen en compte la diferencies de les longituds dels gens, punt que si tenen en compte els valors de RPKM i FPKM.

Els CPM i els log-CPM es poden calcular només a partir de la matriu d'expressió i són suficients per al tipus de comparacions en que estem interessats.

Suposar que no hi ha diferències en l'ús d'isoformes entre condicions, implica que a nivell d'expressió diferencial les anàlisis només s'adrecen a mesurar canvis d'expressió gènica entre les condicions, en lloc de comparar l'expressió entre diversos gens o extreure conclusions sobre nivells d'expressió absoluts. És a dir, les longituds dels gens es mantenen constants per a les comparacions d'interès i de qualsevol diferència observada és el resultat de canvis de condició més que de canvis en la longitud del gen.

Aquí els `counts` en brut es converteixen en valors de CPM i log-CPM mitjançant la funció `cpm` en `edgeR`, les transformacions log utilitzeu un `pseudocount`de 0,25 per evitar un possible valor de zero. Els valors RPKM es calculen tan fàcilment com a valors de CPM utilitzant la funció `rpkm` en edgeR si hi ha disponibles longituds de gens.

```
cpm <- cpm(x)
lcpm <- cpm(x, log = TRUE)
```

Filtrar gens molt poc expressats

Tots els datasets inclouen una barreja de gens que s'expressen i els que no s'expressen. Si bé te interès examinar els gens que s'expressen en una condició però no en una altra, alguns gens no estan expressats al llarg de totes les mostres. De fet, el 23% dels gens d'aquest conjunt de dades tenen zero `counts` en les nou mostres.

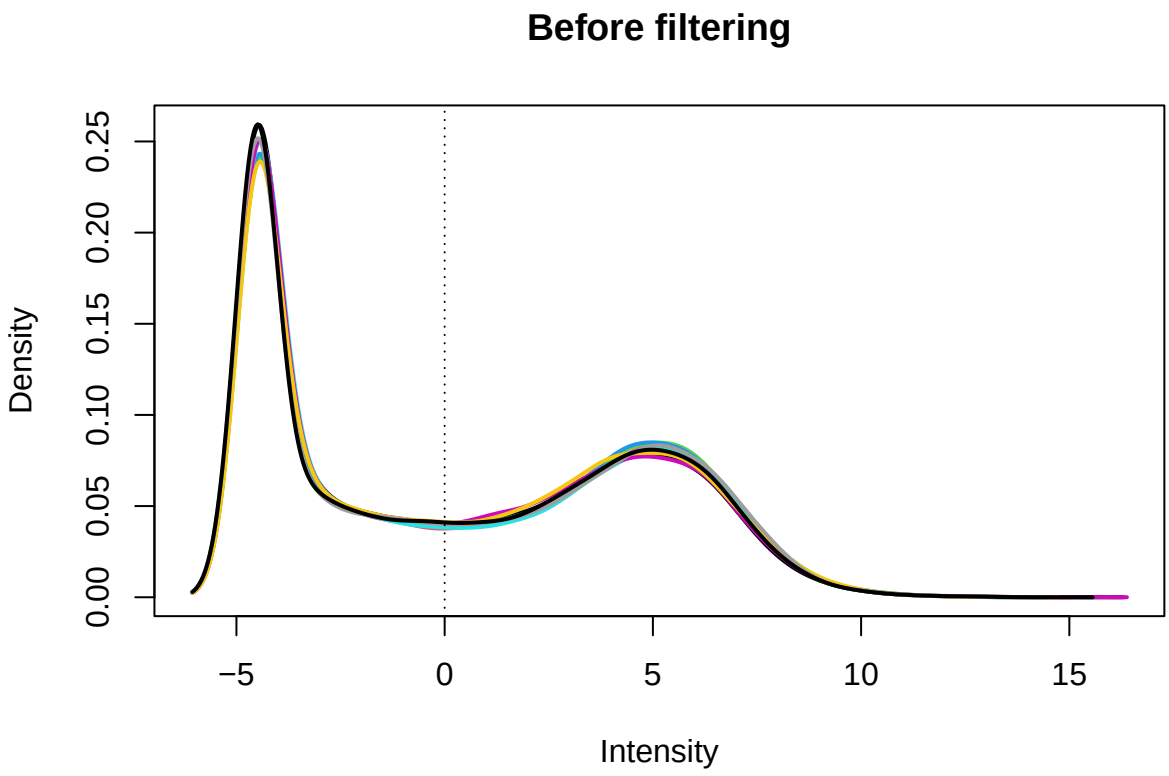
```
table(rowSums(x$counts == 0) == 9)
```

```
##
## FALSE  TRUE
## 22026  5153
```

S'haurien de descartar els gens que no s'expressin a nivell biològic en cap condició, per reduir el subconjunt de gens a aquells que són d'interès i reduir el nombre de proves estadístiques realitzades quan es busca la expressió diferencial.

Examinant els valors de log-CPM, es pot veure que hi ha una gran proporció de gens dins cada mostra que no estan expressats o estan poc expressats.

```
# Visualize distribution of gene expression levels

plotDensities(lcpm, legend = FALSE, main = "Before filtering")
abline(v = 0, lty = 3)
```



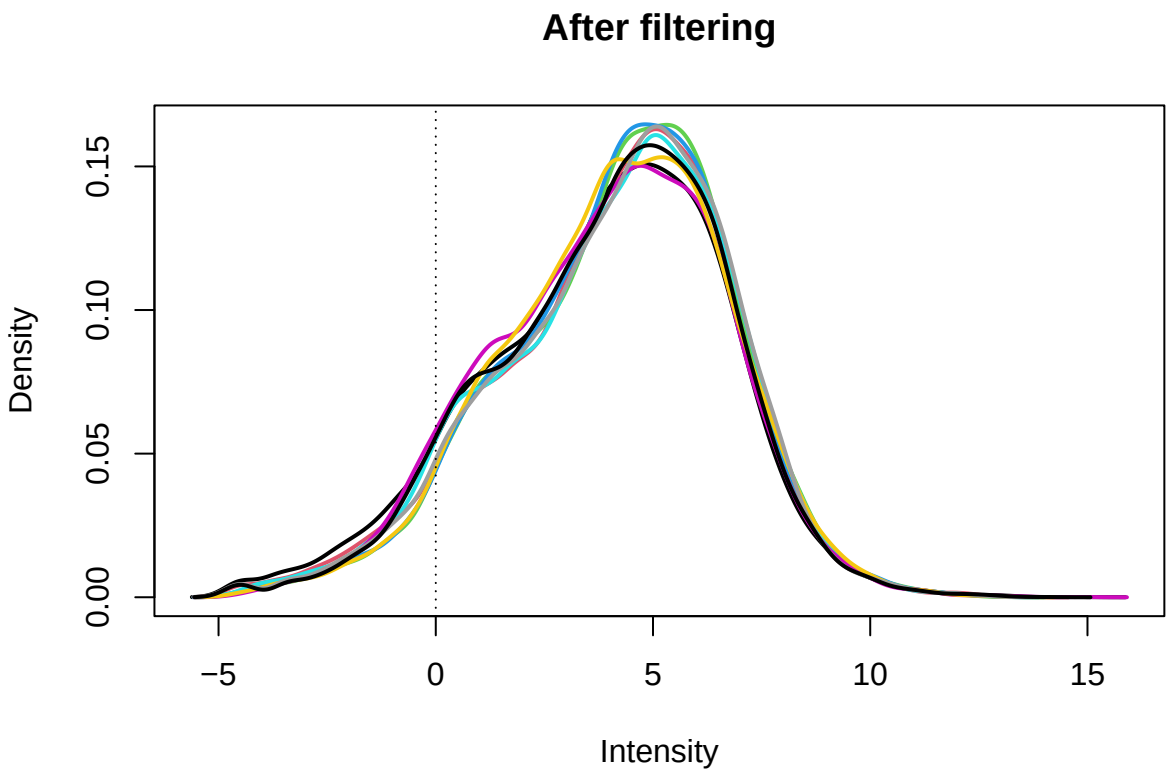
Es considera que els gens s'expressen si el seu valor CPM estar per sobre d'un llindar (utilitzem un valor CPM nominal d'1) i no s'expressen en altre cas.

Un valor CPM d'1 equival a un valor log-CPM de 0. Els gens s'han d'expressar en almenys un grup (o en almenys en tres mostres en tot l'experiment, on es va triar tres, ja que és la mida del grup més reduït) per tal que es mantingui en l'anàlisi.

```
# Only keep genes which have cpm greater than 1 in at least 3 samples.

keep.exprs <- rowSums(cpm > 1) >= 3
x <- x[keep.exprs, , keep.lib.sizes = FALSE]
dim(x)
```

```
## [1] 14165     9
```

```
lcpm <- cpm(x, log=TRUE)
plotDensities(lcpm, legend = FALSE, main = "After filtering")
abline(v = 0, lty = 3)
```



Tot i que es pot utilitzar qualsevol valor prou raonat com a llindar per definir l'expressió. Un valor CPM d'1 per separar els gens expressats dels no expressats funciona bé per a la majoria de conjunts de dades. Aquí, un valor de CPM d'1 significa que a el gen està "expressat" si té almenys 20 `counts` a la mostra amb la profunditat de seqüenciació més baixa (JMS9-P8c, mida de la biblioteca ~20 milions) o almenys 76 `counts` a la mostra amb major profunditat de seqüenciació (JMS8-3, mida de la biblioteca de ~76 milions). Si els `reads`de seqüències es limiten a exons en lloc de gens i/o els experiments tenen una profunda seqüenciació baixa, es pot reduir el llindar.

Utilitzant aquest criteri, el nombre de gens es redueix a aproximadament la meitat del nombre que vam començar amb (14.165 gens). Tingueu en compte que la eliminació afectarà tot l'objecte DGEList, elimina tant els `counts` com els altres elements d'informació.

Normalització de l'expressió dels gens

Durant el procés de preparació o seqüenciació de mostres, factors externs que no siguin d'interès biològic poden afectar l'expressió de mostres individuals. Per exemple, les mostres processades en el primer lot d'un experiment poden tenir major expressió global en comparació amb mostres processades en un segon lot. Se suposa que totes les mostres haurien de tenir un rang i distribució similars dels valors d'expressió. La normalització és necessària per garantir que les distribucions d'expressions de cada mostra són similars en tot l'experiment.

Qualsevol gràfic que mostri les distribucions d'expressió per mostra, com ara els plots de densitat o boxplots, és util per determinar si les mostres són diferents. Observa que la distribució dels valors de log-CPM és similar a totes les mostres dins d'aquest conjunt de dades (density plot anterior).

No obstant així, la normalització mitjançant el mètode `trimmed mean of M-values` (TMM) es realitza mitjançant la funció `calcNormFactors` funcionen en `edgeR`. Els factors de normalització aquí calculats s'utilitzen com a factor per escalar les mides de la biblioteques (Robinson i Oshlack (2010)). Quan es treballa amb objec-

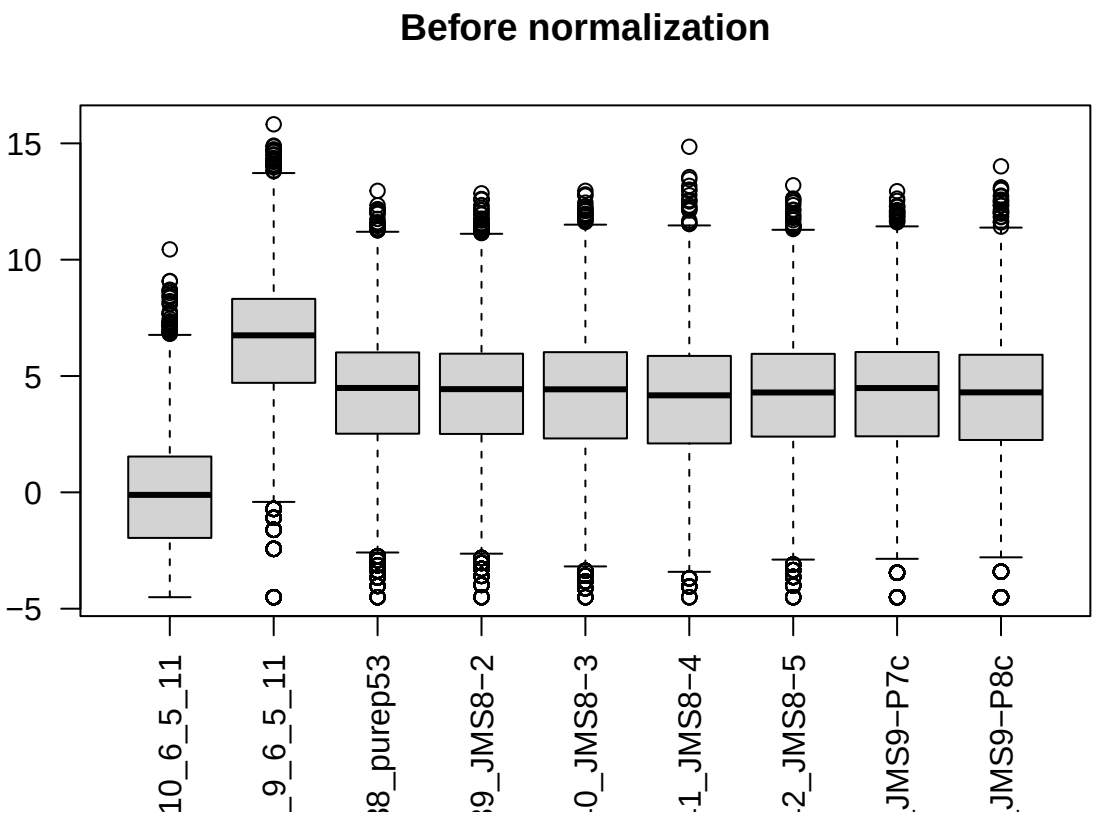
tes DGEList, aquests factors de normalització s'emmagatzemen automàticament `x$samples$norm.factors`. Per a aquest conjunt de dades, l'efecte de la normalització TMM és suau, com és evident en la magnitud dels factors reescalat, que són relativament propers a 1.

```
x <- calcNormFactors(x, method = "TMM")
x$samples$norm.factors
```

```
## [1] 0.8957309 1.0349196 1.0439552 1.0405040 1.0323599 0.9223424 0.9836603
## [8] 1.0827381 0.9792607
```

Per obtenir una millor representació visual dels efectes de la normalització, les dades es van duplicar i es van ajustar de manera que els `counts` de la primera mostra es redueixen al 5% dels seus valors originals i, a la segona mostra, s'inflen 5 vegades més.

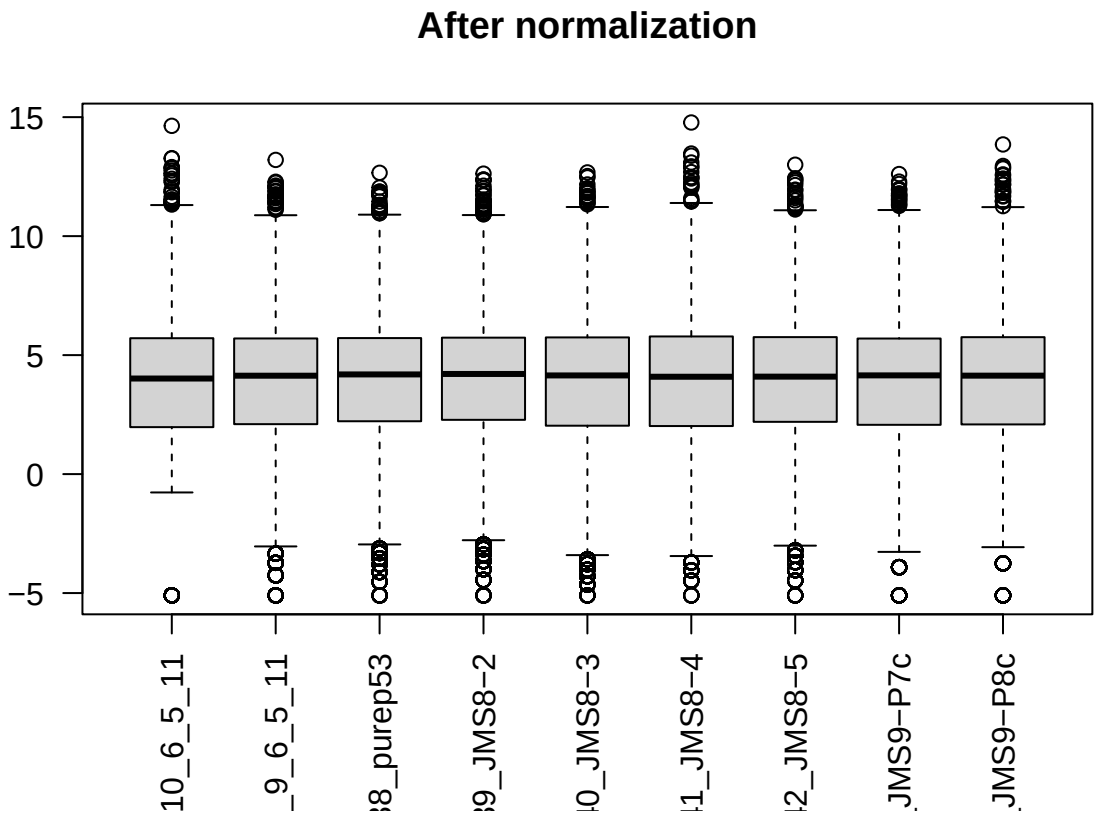
```
# But here is a extreme toy example that demonstrates it will work if
# necessary.

x2 <- x
x2$samples$norm.factors <- 1
x2$counts[,1] <- ceiling(x2$counts[, 1] * 0.05)
x2$counts[,2] <- x2$counts[, 2] * 5

lcpm2 <- cpm(x2, log = TRUE)
boxplot(lcpm2, las = 2, main = "Before normalization")
```




```
x2 <- calcNormFactors(x2)
x2$samples$norm.factors
```

```
## [1] 0.05472223 6.13059440 1.22927355 1.17051887 1.21487709 1.05622968 1.14587663
## [8] 1.26129350 1.11702264
```

```
lcpm2 <- cpm(x2, log = TRUE)
boxplot(lcpm2, las=2, main = "After normalization")
```



El gràfic presenta la distribució d'expressions de les mostres per a dades no normalitzades i normalitzades, on les distribucions son sensiblement diferents en la no normalitzada i similars una vegada la normalització. Aquí la primera mostra té un factor d'escala petit de 0.05, mentre que la segona mostra té un gran factor d'escala de 6.13: els dos valors no s'aproximen a 1.

Representació de les mostres

Segons la nostra opinió, una de les gràfiques exploratòries més importants per examinar les anàlisis d'expressió gènica és el `multidimensional scaling` (MDS).

El MDS mostra similituds i diferències entre mostres en una manera no supervisada perquè es pugui tenir una idea de fins a quin punt es pot detectar una expressió diferencial abans de realitzar els tests gen a gen.

Idealment, voldríem observar que les mostres s'agrupessin bé dins de la condició d'interès principal i es pugués identificar qualsevol mostra que s'allunyés del seu grup i fer-ne el seguiment de fonts d'error o de variació addicional. Si hi ha rèpliques tècniques haurien d'estar molt a prop unes de les altres.

Aquest tipus de representació és pot fer en `limma` mitjançant la funció `plotMDS`. La primera dimensió representa la condició
que millor separa les mostres i explica la major proporció de variació en les dades, les posteriors dimensions expliquen menys variació i són ortogonals entre elles. Quan es tracta d'un disseny experimental amb múltiples factors, es recomana examinar cada factor en diverses dimensions. Si es detecta un clúster de mostres associat

a un dels factors experimentals en qualsevol d'aquestes dimensions, suggereix que el factor contribueix a les diferències d'expressió i val la pena que sigui inclús en el model lineal.

En aquest conjunt de dades, es pot veure que les mostres s'agrupen bé en condicions experimentals en les dimensions 1 i 2, i després els lots de seqüenciació (lane) per la dimensió 3.

Tenint en compte que la primera dimensió explica la proporció més gran de variació en les dades, observeu que el rang de valors es redueix a mesura que passem a dimensions més altes. Mentre que totes les mostres s'agrupen per grups, la diferència transcripcional més gran s'observa entre el basal i el LP, i el basal i el ML en la dimensió 1.

Per aquesta raó, s'espera que es quan fem les comparacions dos a dos entre poblacions cel.lulars tindrem com a resultat un nombre més gran de gens DE per a comparacions amb mostres basals i un nombre relativament reduït de gens DE quan es compara ML amb LP.

Conjunts de dades en que les mostres no s'agrupin per grup experimental poden tenir poca senyal en l'anàlisi d'expressió diferencial.

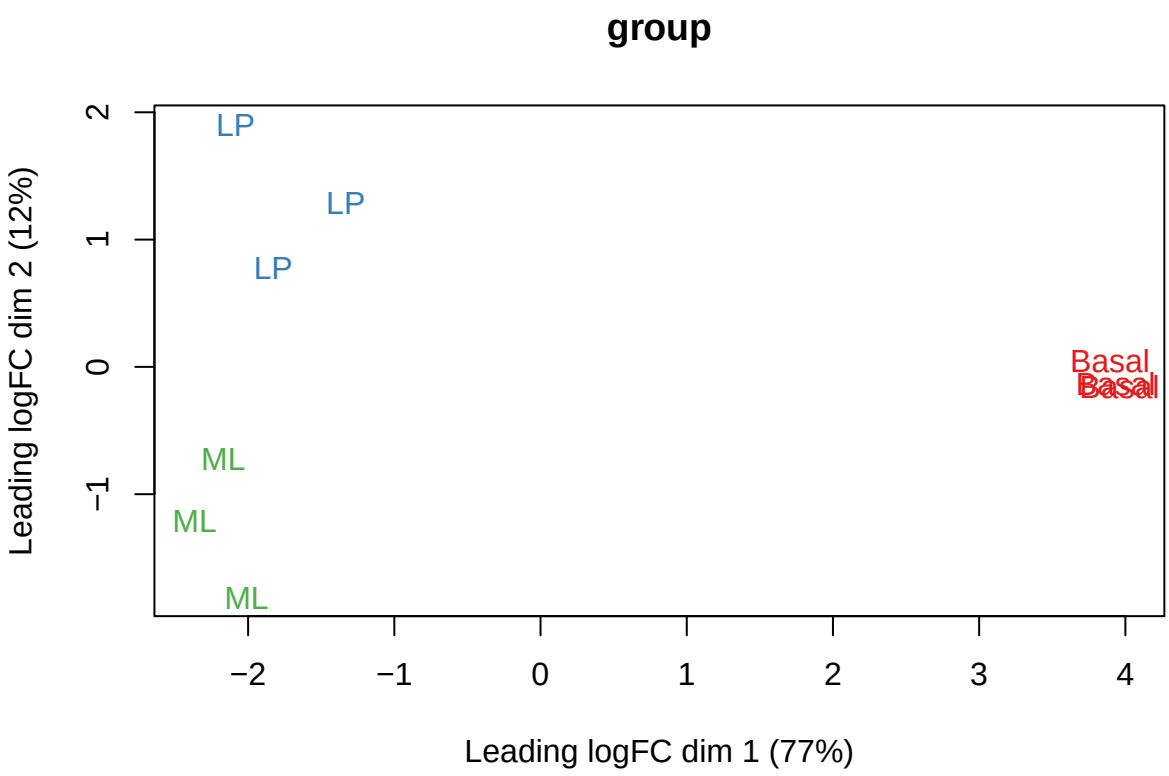
```
# Visualize sample relationships with multidimensional scaling (MDS).

#library("RColorBrewer")
group
```

```
## [1] LP    ML    Basal Basal ML    LP    Basal ML    LP
## Levels: Basal LP ML
```

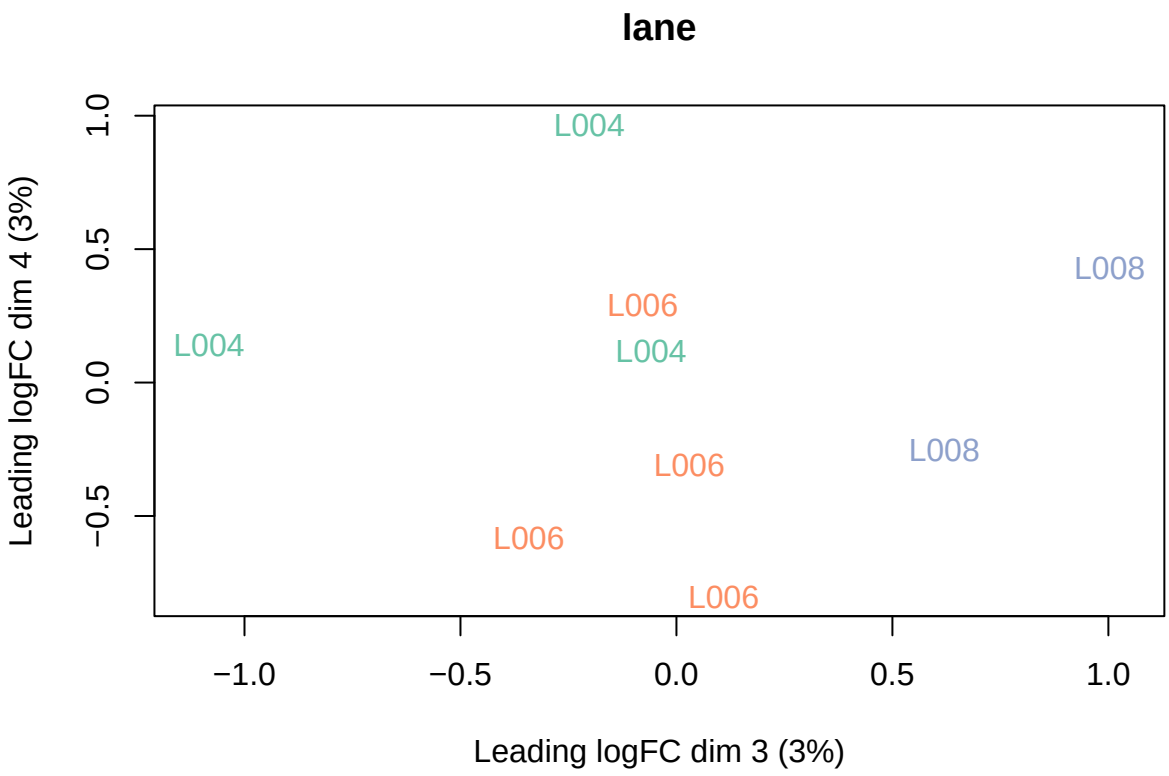
```
col.group <- group
levels(col.group) <-  brewer.pal(nlevels(col.group), "Set1")
col.group <- as.character(col.group)
lane
```

```
## [1] L004 L004 L004 L006 L006 L006 L006 L008 L008
## Levels: L004 L006 L008
```

```
col.lane <- lane
levels(col.lane) <-  brewer.pal(nlevels(col.lane), "Set2")
col.lane <- as.character(col.lane)
plotMDS(lcpm, labels = group, col = col.group,
        main = "group")
```




```
plotMDS(lcpm, labels = lane, col = col.lane, dim = c(3, 4),
        main = "lane")
```



De forma alternativa, el paquet `Glimma` ofereix la comoditat d'un MDS interactiu on es poden explorar diverses dimensions. La funció `glMDSPlot` genera una pàgina html (que s'obre en un navegador si launch = TRUE) amb un MDS a l'esquerre i un barplot que mostra la proporció de variació explicada per cada dimensió, a la dreta panel.

Feu clic a les barres per canviar la parella de dimensions que es representen. Passa el cursor per damunt els punts individuals per veure l'etiqueta de la mostra. L'esquema de colors també es pot canviar per ressaltar la població cel.lular o el lot de seqüenciació (`lane`).

```
## Interactive MDS plot
glMDSPlot(lcpm, labels = paste(group, lane, sep = "_"),
          groups = x$samples[, c(2, 5)], launch = TRUE)
```

Anàlisis d'expressió diferencial

En aquest estudi te interès veure quins gens s'expressen a diferents nivells entre les tres poblacions cel.lulars. En el nostre anàlisi, els models lineals s'ajusten a les dades amb l'assumpció que les dades segueixen una distribució normal. Per començar, s'estableix una matriu de disseny amb informació tant de la població cel.lular com del lot (`lane`).

El model lineal que construím no tindrà un intercept. Així es coneix com la parametrització de `group-means` per l'usuari de `limma`. L'avantatge de tenir en cada coeficient (beta) el nivell d'expressió mitjà d'aquest grup és que el fa més senzill per fer proves d'hipòtesis específiques.

```
# Construct linear model ---------------------------------------------
design <- model.matrix(~0 + group + lane)
colnames(design) <- gsub("group", "", colnames(design))
```

```
design
```

```
##   Basal LP ML laneL006 laneL008
## 1     0  1  0        0        0
## 2     0  0  1        0        0
## 3     1  0  0        0        0
## 4     1  0  0        1        0
## 5     0  0  1        1        0
## 6     0  1  0        1        0
## 7     1  0  0        1        0
## 8     0  0  1        0        1
## 9     0  1  0        0        1
## attr(,"assign")
## [1] 1 1 1 2 2
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
## attr(,"contrasts")$lane
## [1] "contr.treatment"
```

Els contrastos per fer les comparacions entre parelles de poblacions cel.lulars es configuren a `limma` mitjançant la funció `makeContrasts`.

```
contr.matrix <- makeContrasts(BasalvsLP = Basal - LP,
                              BasalvsML = Basal - ML,
                              LPvsML = LP - ML,
                              levels = colnames(design))
contr.matrix
```

```
##           Contrasts
## Levels     BasalvsLP BasalvsML LPvsML
##   Basal            1         1      0
##   LP              -1         0      1
##   ML               0        -1     -1
##   laneL006         0         0      0
##   laneL008         0         0      0
```

Convertir els counts per utilitzar-los en el model lineal

S'ha demostrat que els `counts` en RNAseq, la variància no és independent de la mitjana, així és cert tant pels `counts` en brut com en els log-CPM.

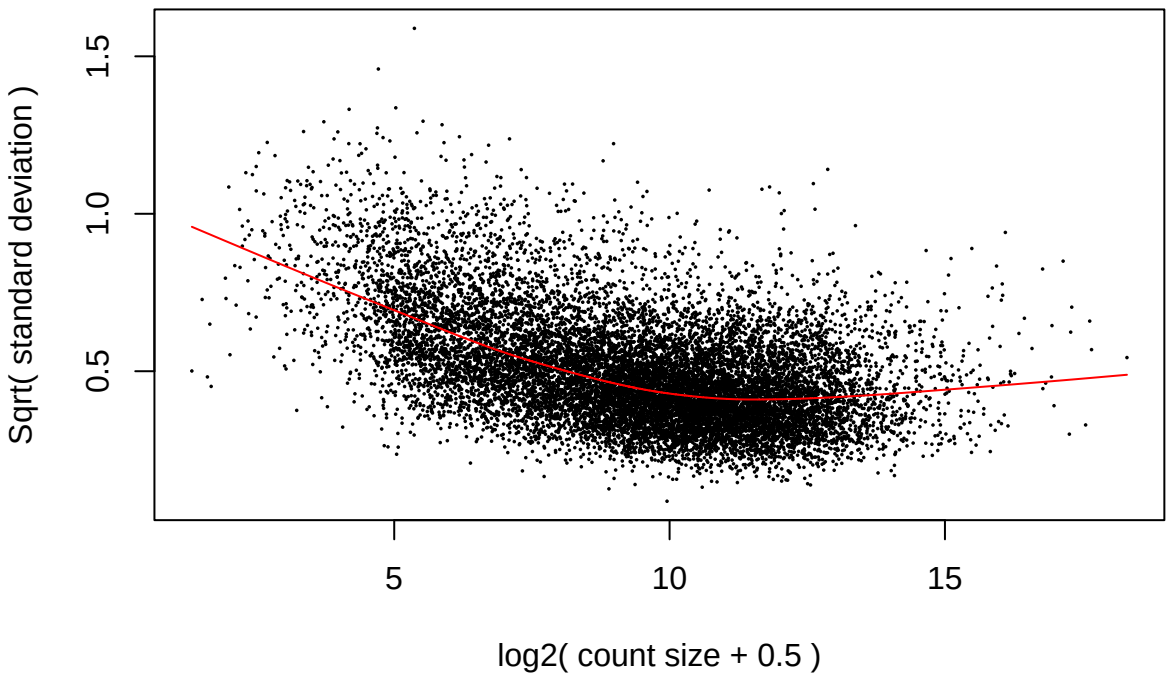
Mètodes que modelen els `counts` mitjançant una distribució binomial negativa assumeix una relació quadràtica entre variància i mitjana.

En limma, assumeix que els log-CPM es distribueixen normalment i la relació entre mitjana-variància es realitza mitjançant uns pesos calculats per la funció `voom`.

Quan opera en un objecte `DGEList`, `voom` converteix els `counts` en valors en log-CPM mitjançant l'extracció automàtica de les mides de biblioteca i factors de normalització de la mateixa x.

La relació mitjana-variància dels valors log-CPM d'aquest conjunt de dades es mostra a continuació.

```
v <- voom(x, design, plot=TRUE)
```




```
v
```

```
## An object of class "EList"
## $genes
##    ENTREZID SYMBOL TXCHROM
## 1    497097   Xkr4    chr1
## 6     27395 Mrpl15    chr1
## 7     18777 Lypla1    chr1
## 9     21399  Tcea1    chr1
## 10    58175  Rgs20    chr1
## 14160 more rows ...
##
## $targets
##                                                          files group lib.size
## 1/GSM1545535_10_6_5_11 dadesRNAseq1/GSM1545535_10_6_5_11.txt    LP 29409426
## 1/GSM1545536_9_6_5_11   dadesRNAseq1/GSM1545536_9_6_5_11.txt    ML 36528591
## 1/GSM1545538_purep53     dadesRNAseq1/GSM1545538_purep53.txt Basal 59598629
## 1/GSM1545539_JMS8-2       dadesRNAseq1/GSM1545539_JMS8-2.txt Basal 53382070
## 1/GSM1545540_JMS8-3       dadesRNAseq1/GSM1545540_JMS8-3.txt    ML 78175314
## 1/GSM1545541_JMS8-4       dadesRNAseq1/GSM1545541_JMS8-4.txt    LP 55762781
## 1/GSM1545542_JMS8-5       dadesRNAseq1/GSM1545542_JMS8-5.txt Basal 54115150
## 1/GSM1545544_JMS9-P7c   dadesRNAseq1/GSM1545544_JMS9-P7c.txt    ML 23043111
## 1/GSM1545545_JMS9-P8c   dadesRNAseq1/GSM1545545_JMS9-P8c.txt    LP 19525423
##                        norm.factors lane
## 1/GSM1545535_10_6_5_11    0.8957309 L004
## 1/GSM1545536_9_6_5_11     1.0349196 L004
## 1/GSM1545538_purep53      1.0439552 L004
```

```
## 1/GSM1545539_JMS8-2       1.0405040 L006
## 1/GSM1545540_JMS8-3       1.0323599 L006
## 1/GSM1545541_JMS8-4       0.9223424 L006
## 1/GSM1545542_JMS8-5       0.9836603 L006
## 1/GSM1545544_JMS9-P7c     1.0827381 L008
## 1/GSM1545545_JMS9-P8c     0.9792607 L008
##
## $E
##         Samples
## Tags     1/GSM1545535_10_6_5_11 1/GSM1545536_9_6_5_11 1/GSM1545538_purep53
##   497097              -4.293244             -3.869026             2.522753
##   27395                3.875010              4.400568             4.521172
##   18777                4.707695              5.559334             5.400569
##   21399                4.784462              4.741999             5.374548
##   58175                3.943567              3.294875            -1.767924
##         Samples
## Tags     1/GSM1545539_JMS8-2 1/GSM1545540_JMS8-3 1/GSM1545541_JMS8-4
##   497097            3.302006           -4.481286           -3.993876
##   27395             4.570624            4.322845            3.786547
##   18777             5.171235            5.627798            5.081794
##   21399             5.130925            4.848030            4.944024
##   58175            -1.880302            2.993289            3.357379
##         Samples
## Tags     1/GSM1545542_JMS8-5 1/GSM1545544_JMS9-P7c 1/GSM1545545_JMS9-P8c
##   497097            3.306782             -3.204336             -5.287282
##   27395             3.918878              4.345642              4.132678
##   18777             5.080061              5.757404              5.150470
##   21399             5.158292              5.036933              4.987679
##   58175            -2.114104              3.142621              3.523290
## 14160 more rows ...
##
## $weights
##           [,1]      [,2]      [,3]     [,4]      [,5]      [,6]      [,7]
## [1,]  1.183974  1.183974 20.526779 20.97747  1.773562  1.217142 21.125740
## [2,] 20.879554 26.561871 31.596323 29.66102 32.558344 26.745293 29.792090
## [3,] 28.003202 33.695540 34.845507 34.45673 35.148529 33.550527 34.517259
## [4,] 27.670233 29.595778 34.901302 34.43298 34.841349 33.159425 34.493456
## [5,] 19.737381 18.658333  3.184207  2.62986 24.191635 24.014937  2.648747
##           [,8]      [,9]
## [1,]  1.183974  1.183974
## [2,] 21.900102 17.150677
## [3,] 31.440457 25.228325
## [4,] 26.136796 24.502247
## [5,] 13.149278 14.351930
## 14160 more rows ...
##
## $design
##   Basal LP ML laneL006 laneL008
## 1     0  1  0        0        0
## 2     0  0  1        0        0
## 3     1  0  0        0        0
## 4     1  0  0        1        0
## 5     0  0  1        1        0
## 6     0  1  0        1        0
```

```
## 7     1  0  0        1        0
## 8     0  0  1        0        1
## 9     0  1  0        0        1
## attr(,"assign")
## [1] 1 1 1 2 2
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
## attr(,"contrasts")$lane
## [1] "contr.treatment"
```

Usualment, el `voom-plot` mostra una tendència decreixent entre les mitjans i les variàncies que resulta d'una combinació de variació técnica en l'experiment de seqüenciació i la variació biològica entre les mostres replicades de diferents poblacions cel.lulars.

Els experiments amb alta variació biològica solen donar lloc a tendències més planes, on els valors de la variància s'aplanen a la part alta valors d'expressió. Els experiments amb baixa variació biològica solen donar lloc a tendències decreixents brusques.

D'altra banda, el voom-plot ofereix una comprovació visual del filtratge realitzat anteriorment. Si el filtrat de gens poc expressats ha estat insuficient, es pot observar una baixada dels nivells de variància a la part baix de l'expressió a causa de recomptes molt reduïts. Si s'observa així, cal tornar al pas de filtratge anterior i augmentar-ne llindar d'expressió aplicat al conjunt de dades.

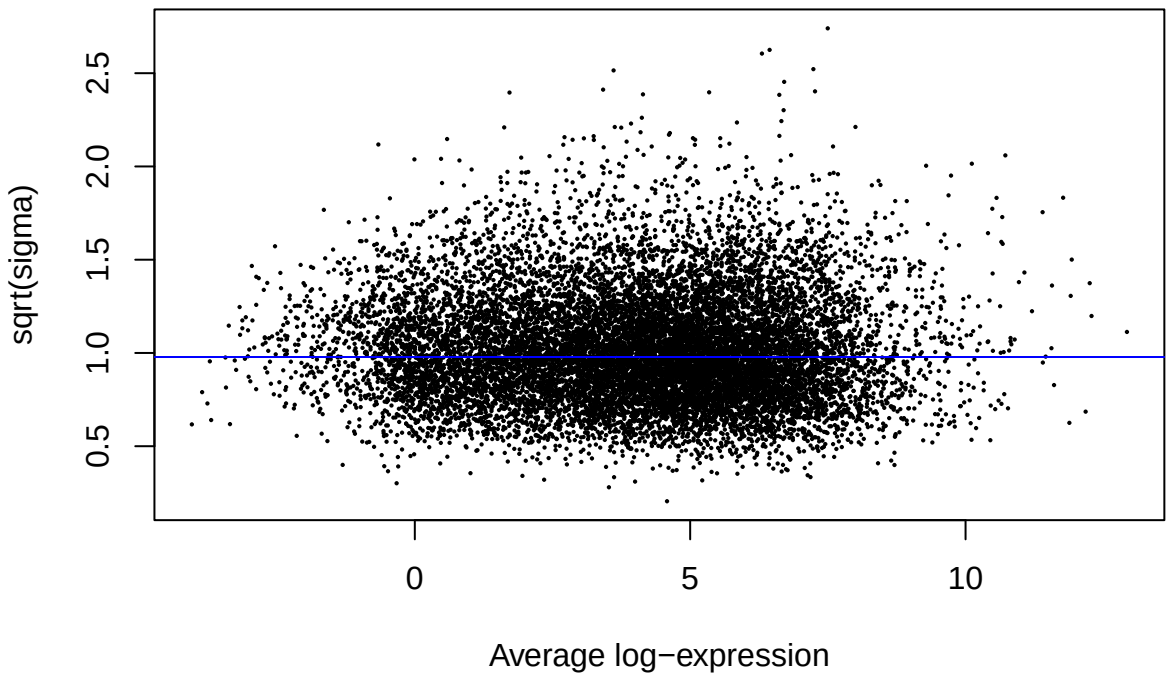
Observa que que els altres slots de dades emmagatzemats a l'objecte DGEList que contenen informació a nivell de gen i de mostra són conservat a l'objecte `v` creat per `voom`. El data frame `v$genes` es equivalent a `x$genes`, `v$targets` equival a `x$samplesi` els valors d'expressió emmagatzemats a `v$E` són analegs als `x$counts`, encara que a escala transformada. A més, l'objecte voom té una matriu de pesos de precisió `v$weights` i la matriu de disseny en `v$design`.

Ajust del model lineal i comparacions

El model lineal en limma es realitza mitjançant les funcions `lmFit` i `contrasts.fit` escrites originalment per a aplicació en microarrays. Les funcions es poden utilitzar tant per a dades de microarray com RNA-seq i ajusten un model independent per a l'expressió de cada gen. Incorpora un modelat Bayesià empíric que te present la informació de tots els gens per obtenir estimacions més precises de variabilitat a nivell de cada gen.

Les variàncies residuals es representen en el gràfic seguent front a les expressions mitjanes. Del gràfic es pot veure que la variància ja no depén a nivell d'expressió mitjana (model homocedastic).

```
vfit <- lmFit(v, design)
vfit <- contrasts.fit(vfit, contrasts=contr.matrix)
efit <- eBayes(vfit)
plotSA(efit)
```



Examinar l'expressió diferencial

Per fer una visió ràpida als nivells d'expressió diferencials, es pot presentar una taula amb el nombre de gens significativament up-regulats i down-regulats. La significació es defineix mitjançant p-valor ajustat que s'estableix per defecte en un 5%.

En la comparació dels nivells d'expressió entre basal i LP, es troben 4127 gens down-regulats en relació a LP i 4298 gens estan up-regulats en LP - un total de 8425 gens DE. Un total de 8510 gens DE es troben entre els basals i els ML (4338 down i 4172 up), i un total de 5340 gens DE s'han trobat entre LP i ML (2895 down i 2445 up). El nombre més gran de gens DE observats per a les comparacions entre la població basal són coherents amb el que vam observar en la representació MDS.

```
# Tabulate the results
summary(decideTests(efit))
```

```
##        BasalvsLP BasalvsML LPvsML
## Down        4127      4338   2895
## NotSig      5740      5655   8825
## Up          4298      4172   2445
```

La mida de l'efecte (log-fold-change) és important en l'anàlisi (per exemple, es vol prioritzar certs gens entre altres) es pot especificar un logFC minim amb la funció `treat`. Per exemple, un log-fold-change de 1 equival a 2 vegades la diferencia entre tipus cel · lulars en escala original. El nombre de gens DE es redueixen a un total de 3135 gens entre basals versus LP, 3270 DE gens entre basals versus ML i 385 DE gens entre LP vers ML

```
tfit <- treat(vfit, lfc = 1)
dt <- decideTests(tfit)
```

```
summary(dt)
```

```
##        BasalvsLP BasalvsML LPvsML
## Down        1417      1512    203
## NotSig     11030     10895  13780
## Up          1718      1758    182
```

```
# Create a venn diagram of the results.
head(dt)
```

```
## TestResults matrix
##         Contrasts
##          BasalvsLP BasalvsML LPvsML
##   497097         1         1      0
##   27395          0         0      0
##   18777          0         0      0
##   21399          0         0      0
##   58175         -1        -1      0
##   108664         0         0      0
```

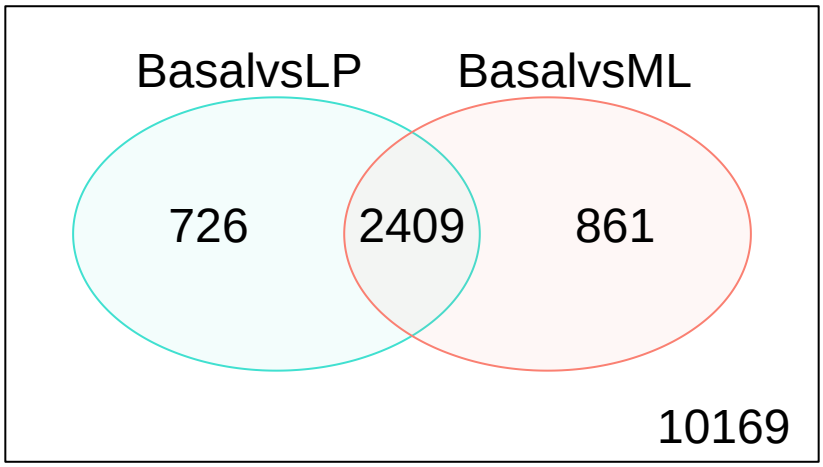
```
de.common <- which(dt[, 1] != 0 & dt[, 2] != 0)
length(de.common)
```

```
## [1] 2409
```

```
head(tfit$genes$SYMBOL[de.common], n = 20)
```

```
##  [1] "Xkr4"     "Rgs20"    "Cpa6"     "Sulf1"    "Eya1"     "Msc"
##  [7] "Sbspon"   "Pi15"     "Crispld1" "Kcnq5"    "Ptpn18"   "Arhgef4"
## [13] "Cracdl"   "Aff3"     "Npas2"    "Tbc1d8"   "Creg2"    "Il1r1"
## [19] "Il18r1"   "Il18rap"
```

```
vennDiagram(dt[, 1:2], circle.col = c("turquoise", "salmon"))
```




```
# Save results
#write.fit(tfit, dt, file = "results.txt")
```

Identificar els gens top diferencialment expressats

Els gens DE mès destacats es poden llistar amb `TopTreat` en el cas de `treat` (o `TopTable` en el cas de `eBayes`).

De manera predeterminada, TopTreat organitza gens de menor p-valor ajustat amb la informació de gen associat, log-FC, log-CPM promig, estadístic t, p-valor raw i p-valor ajustat de cada gen. El nombre de gens llistats es pot especificar, on `n = Inf` inclou tots els gens. Els gens Cldn7 i Rasef són un dels principals gens de DE tant basal vers LP i basal vers ML.

```
# Identify top DE genes.
basal.vs.lp <- topTreat(tfit, coef = 1, n = Inf)
basal.vs.ml <- topTreat(tfit, coef = 2, n = Inf)
head(basal.vs.lp)
```

```
##        ENTREZID SYMBOL TXCHROM     logFC  AveExpr         t      P.Value
## 12759     12759    Clu   chr14 -5.442877 8.857907 -33.44429 3.990899e-10
## 53624     53624  Cldn7   chr11 -5.514605 6.296762 -32.94533 4.503694e-10
## 242505   242505  Rasef    chr4 -5.921741 5.119585 -31.77625 6.063249e-10
## 67451     67451   Pkp2   chr16 -5.724823 4.420495 -30.65370 8.010456e-10
## 228543   228543   Rhov    chr2 -6.253427 5.486640 -29.46244 1.112729e-09
## 70350     70350  Basp1   chr15 -6.073297 5.248349 -28.64890 1.380545e-09
##           adj.P.Val
## 12759  2.703871e-06
```

```
## 53624  2.703871e-06
## 242505 2.703871e-06
## 67451  2.703871e-06
## 228543 2.703871e-06
## 70350  2.703871e-06
```

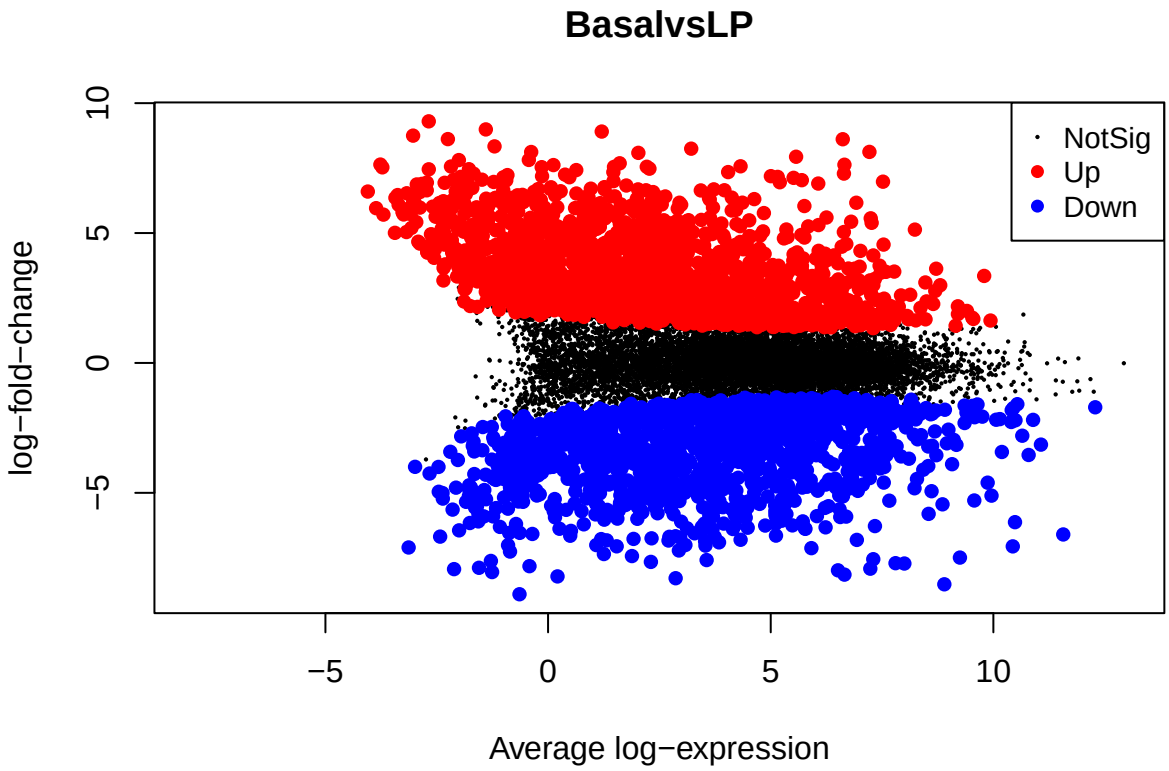
```
head(basal.vs.ml)
```

```
##        ENTREZID  SYMBOL TXCHROM     logFC  AveExpr         t      P.Value
## 242505   242505   Rasef    chr4 -6.510470 5.119585 -35.49093 2.573575e-10
## 53624     53624   Cldn7   chr11 -5.469160 6.296762 -32.52520 4.978446e-10
## 12521     12521    Cd82    chr2 -4.667737 7.070963 -31.82187 5.796191e-10
## 71740     71740 Nectin4    chr1 -5.556046 5.166292 -31.29987 6.760578e-10
## 20661     20661   Sort1    chr3 -4.908119 6.705784 -31.23083 6.761331e-10
## 15375     15375   Foxa1   chr12 -5.753884 5.625064 -28.34612 1.487280e-09
##           adj.P.Val
## 242505 1.915485e-06
## 53624  1.915485e-06
## 12521  1.915485e-06
## 71740  1.915485e-06
## 20661  1.915485e-06
## 15375  2.281914e-06
```

Visualització de l'expressio diferencial

Per resumir els resultats de tots els gens de forma visual podem representar el log-FC sobre la mitjana de valors log-CPM mitjançant la funció plotMD.

```
# Visualize DE genes.
plotMD(tfit, column = 1, status = dt[, 1], main = colnames(tfit)[1],
       xlim = c(-8, 13))
```



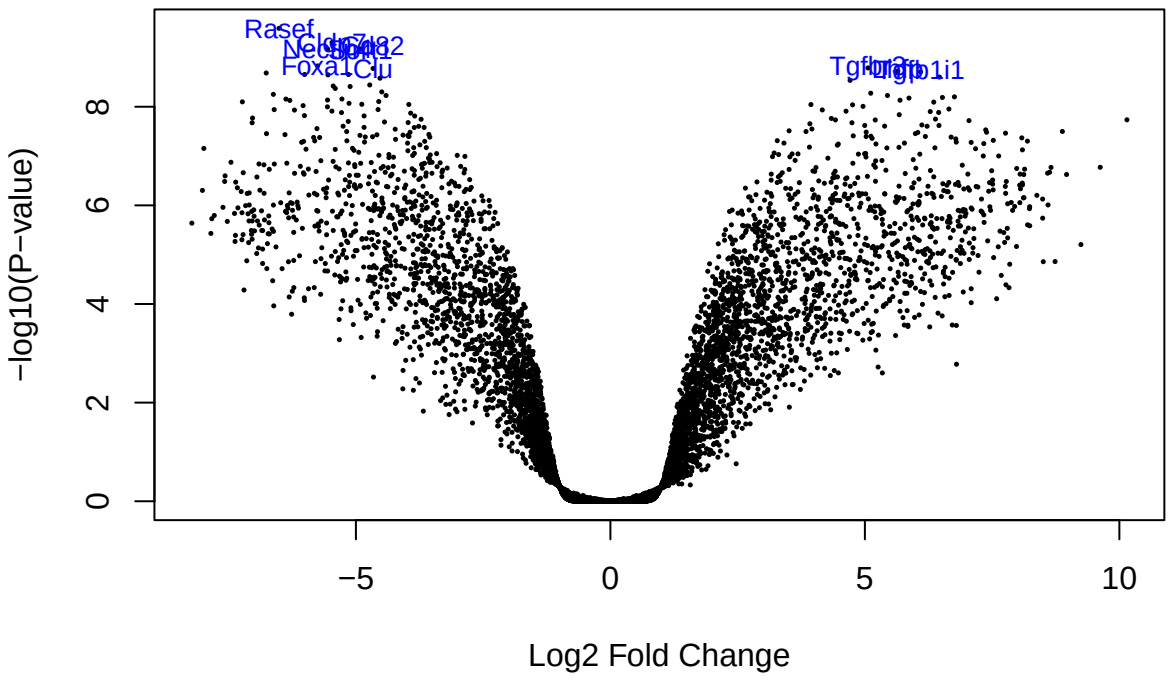
`Glimma` amplia aquesta funcionalitat proporcionant una representació interactiva mitjançant la funció `glMDPlot`.

```
glMDPlot(tfit, coef = 1, status = dt, main = colnames(tfit)[1],
         side.main = "ENTREZID", counts = x$counts, groups = group,
         launch = TRUE)
```

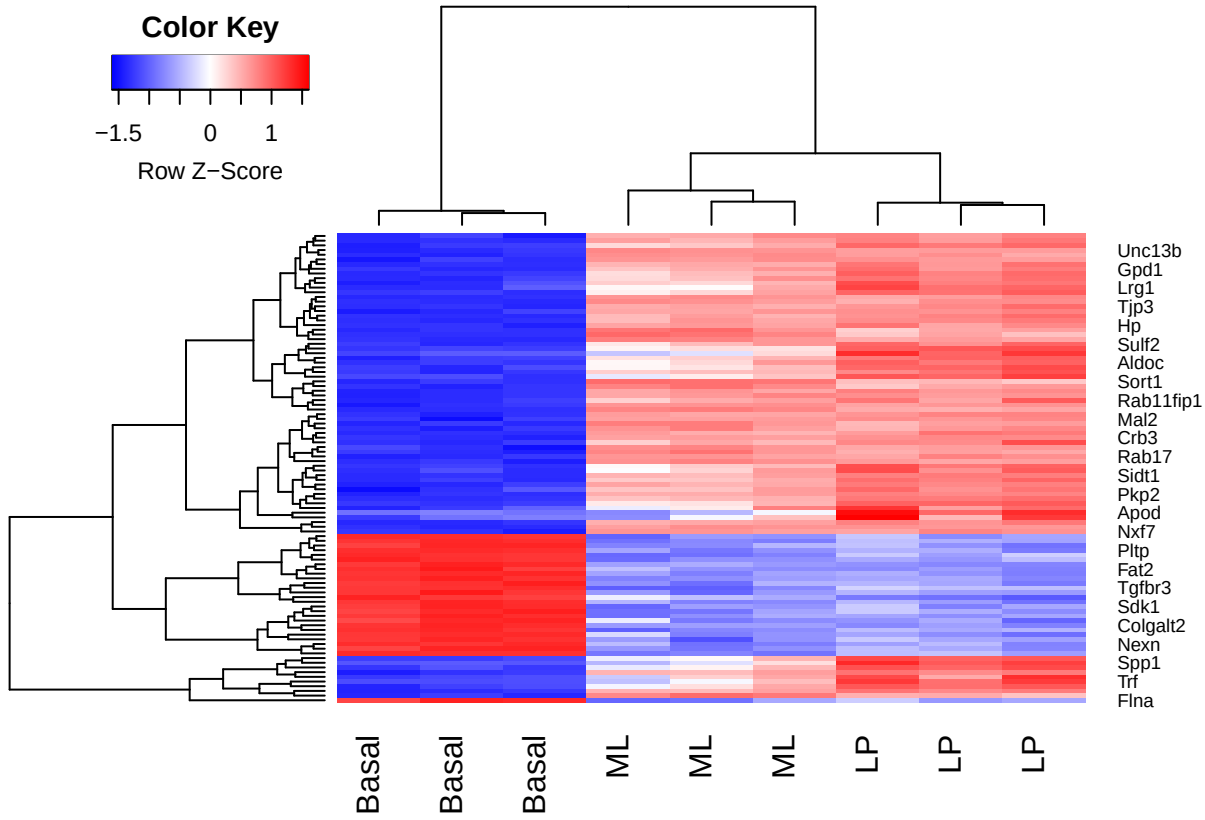
Es crea un mapa de calor (heatmap) pels 100 primers gens de DE (segons el p-valor ajustat) a partir del contrast basal vers el LP, amb la funció `heatmap.2` del paquet `gplots`. El heatmap agrupa correctament les mostres per tipus cel · lulars i reordena els gens en blocs amb patrons d'expressió similars. En el heatmap observem que l'expressió de mostres de ML i LP és molt similar en aquests 100 primers gens amb expressió diferencial entre basal i LP.

També resulta molt útil una representació basada en el volcanoplot.

```
volcanoplot(tfit, coef = 2, style = "p-value", highlight = 10, names = tfit$genes$SYMBOL, hl.col="blue"
```




```
# View heatmap of top 100 DE genes between Basal and LP cells.
basal.vs.lp.topgenes <- basal.vs.lp$ENTREZID[1:100]
i <- which(v$genes$ENTREZID %in% basal.vs.lp.topgenes)
mycol <- colorpanel(1000, "blue", "white", "red")
heatmap.2(v$E[i, ], scale = "row",
          labRow = v$genes$SYMBOL[i], labCol = group,
          col = mycol, trace = "none", density.info = "none")
```



Anàlisi d'enriquiment en conjunts de gens (Gene set testing)

Acabem aquesta anàlisi amb algunes proves d'enriquiment en conjunts de gens aplicant la funció `camera` a les signatures de la col · leccio MSigDB C2 del Broad Institute que han estat definides per a ratolí i que estan disponibles com a objectes Rdata a http://bioinf.wehi.edu.au/software/MSigDB/.

Els conjunts de gens C2 ha estat curat a partir de bases de dades, publicacions i experts, es seleccionen conjunts de gens per representar estats o processos biològics ben definits.

La funció de `camera` realitza una prova per avaluar si els gens d'un conjunt determinat estan altament ordenats en termes d'expressió diferencial respecte dels gens que no es troben en el conjunt. Utilitza el model lineal en `limma`, emptant tant la matriu de disseny com la matriu de contrast (si està present) i els pesos del `voom`.

```
load(url("http://bioinf.wehi.edu.au/software/MSigDB/mouse_c2_v5p1.rdata"))
idx <- ids2indices(Mm.c2,id=rownames(v))
cam.BasalvsLP <- camera(v,idx,design,contrast=contr.matrix[,1])
head(cam.BasalvsLP,5)
```

```
##                                             NGenes Direction       PValue
## LIM_MAMMARY_STEM_CELL_UP                       739        Up 1.134757e-18
## LIM_MAMMARY_STEM_CELL_DN                       630      Down 1.569957e-15
## ROSTY_CERVICAL_CANCER_PROLIFERATION_CLUSTER    163        Up 1.437987e-13
## SOTIRIOU_BREAST_CANCER_GRADE_1_VS_3_UP         183        Up 2.181862e-13
## LIM_MAMMARY_LUMINAL_PROGENITOR_UP               87      Down 6.734613e-13
##                                                    FDR
## LIM_MAMMARY_STEM_CELL_UP                  5.360590e-15
## LIM_MAMMARY_STEM_CELL_DN                  3.708238e-12
```

```
## ROSTY_CERVICAL_CANCER_PROLIFERATION_CLUSTER 2.264351e-10
## SOTIRIOU_BREAST_CANCER_GRADE_1_VS_3_UP      2.576779e-10
## LIM_MAMMARY_LUMINAL_PROGENITOR_UP           6.362863e-10
```

```
cam.BasalvsML <- camera(v,idx,design,contrast=contr.matrix[,2])
head(cam.BasalvsML,5)
```

```
##                                           NGenes Direction       PValue
## LIM_MAMMARY_STEM_CELL_UP                     739        Up 5.090937e-23
## LIM_MAMMARY_STEM_CELL_DN                     630      Down 5.132446e-19
## LIM_MAMMARY_LUMINAL_MATURE_DN                166        Up 8.875174e-16
## LIM_MAMMARY_LUMINAL_MATURE_UP                180      Down 6.287301e-13
## ROSTY_CERVICAL_CANCER_PROLIFERATION_CLUSTER   163        Up 1.684323e-12
##                                                    FDR
## LIM_MAMMARY_STEM_CELL_UP                     2.404959e-19
## LIM_MAMMARY_STEM_CELL_DN                     1.212284e-15
## LIM_MAMMARY_LUMINAL_MATURE_DN                1.397544e-12
## LIM_MAMMARY_LUMINAL_MATURE_UP                7.425303e-10
## ROSTY_CERVICAL_CANCER_PROLIFERATION_CLUSTER 1.591348e-09
```

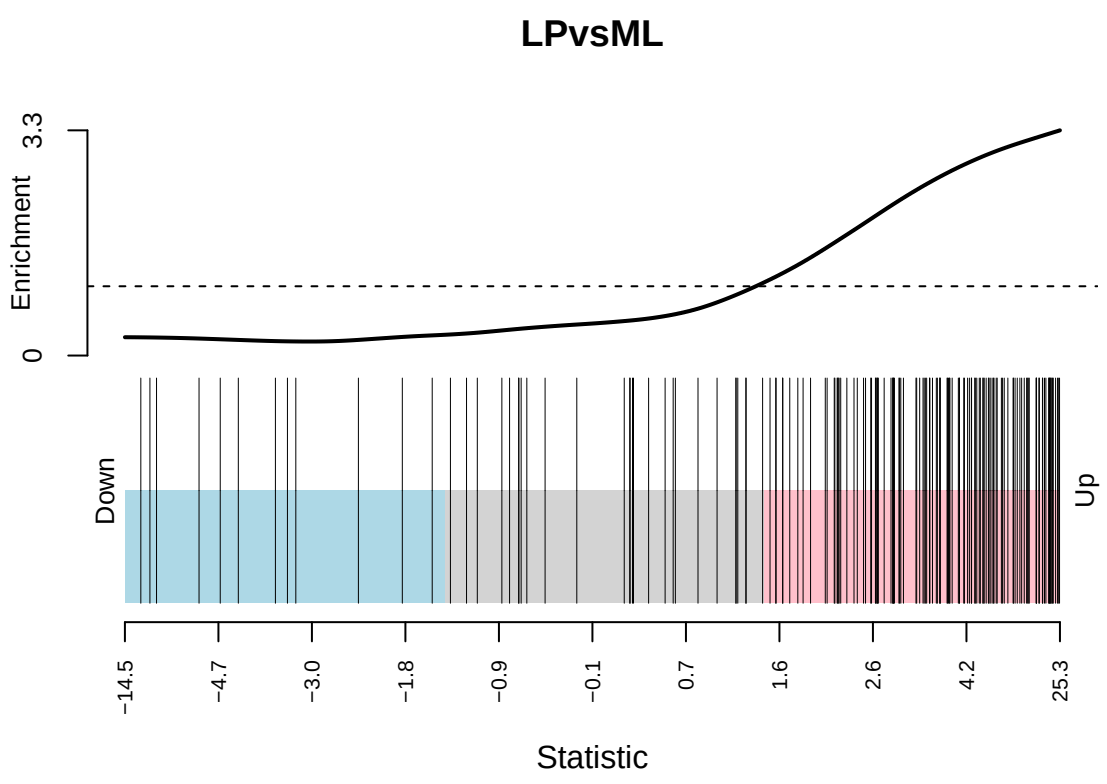
```
cam.BasalvsML <- camera(v,idx,design,contrast=contr.matrix[,3])
head(cam.BasalvsML,5)
```

```
##                                         NGenes Direction       PValue
## LIM_MAMMARY_LUMINAL_MATURE_UP              180      Down 8.497295e-14
## LIM_MAMMARY_LUMINAL_MATURE_DN              166        Up 1.439890e-13
## LIM_MAMMARY_LUMINAL_PROGENITOR_UP           87        Up 3.840915e-11
## REACTOME_RESPIRATORY_ELECTRON_TRANSPORT     91      Down 2.655349e-08
## NABA_CORE_MATRISOME                        222        Up 4.430361e-08
##                                                  FDR
## LIM_MAMMARY_LUMINAL_MATURE_UP           3.401020e-10
## LIM_MAMMARY_LUMINAL_MATURE_DN           3.401020e-10
## LIM_MAMMARY_LUMINAL_PROGENITOR_UP       6.048160e-08
## REACTOME_RESPIRATORY_ELECTRON_TRANSPORT 3.135967e-05
## NABA_CORE_MATRISOME                     4.185805e-05
```

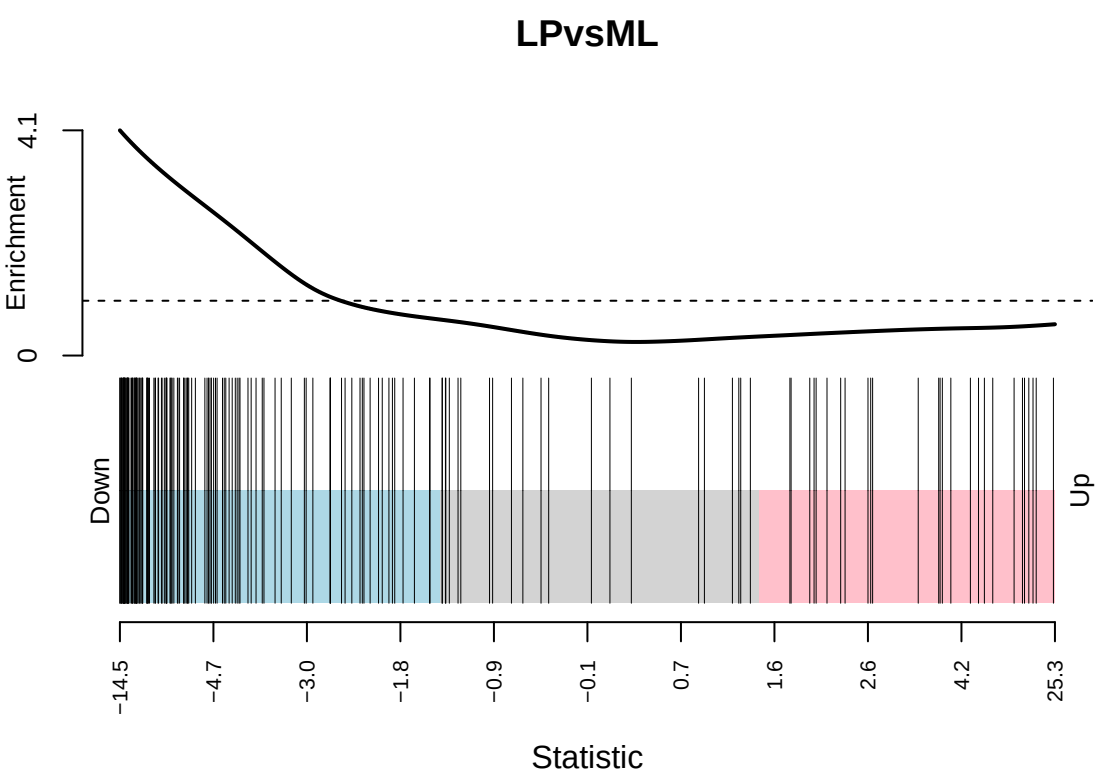
Aquest experiment és l'equivalent al generat per Lim et al. (2010), que van fer servir microarrays Illumina per estudiar les mateixes poblacions de cel.lules, de manera que resulta tranquil · litzador veure les signatures gèniques d'aquesta publicació a la part superior de la llista per a cada contrast.

Fem un barcode de les signatures: "Mature Luminal gene sets (Up and Down) in the LP versus ML contrast". Tingueu en compte que aquests conjunts van en el sentit contrari al nostre conjunt de dades a causa de la nostra parametrització que compara LP amb ML en lloc de l'inrevés (si es revertés el contrast, les indicacions serien consistents).

```
barcodeplot(efit$t[,3], index=idx$LIM_MAMMARY_LUMINAL_MATURE_DN,
 main="LPvsML")
```




```
barcodeplot(efit$t[,3], index=idx$LIM_MAMMARY_LUMINAL_MATURE_UP,
 main="LPvsML")
```



Bibliografia

Robinson, Mark D, i Alicia Oshlack. 2010. «A scaling normalization method for differential expression analysis of RNA-seq data». *Genome biology* 11 (3): 1-9.

Sheridan, Julie M, Matthew E Ritchie, Sarah A Best, Kun Jiang, Tamara J Beck, François Vaillant, Kevin Liu, et al. 2015. «A pooled shRNA screen for regulators of primary mammary stem and progenitor cells identifies roles for Asap1 and Prox1». *BMC cancer* 15: 1-13.