

# BB512/BB612 - Week IX

```
suppressPackageStartupMessages(library(limma))
suppressPackageStartupMessages(library(edgeR))
suppressPackageStartupMessages(library(Mus.musculus))
```

## Microarray

We'll analyze GSE73577 "Gene expression profiles in nineteen Kawasaki Disease (KD) patients before and after intravenous immunoglobulin (IVIG) treatment".

```
# Version info: R 3.2.3, Biobase 2.30.0, GEOquery 2.40.0, limma 3.26.8
#####
# Differential expression analysis with limma
library(GEOquery)

## Setting options('download.file.method.GEOquery'='auto')
## Setting options('GEOquery.inmemory.gpl'=FALSE)
library(limma)
library(umap)

# load series and platform data from GEO

gset <- getGEO("GSE73577", GSEMatrix =TRUE, AnnotGPL=TRUE)

## Found 1 file(s)

## GSE73577_series_matrix.txt.gz

## Rows: 45015 Columns: 39
## -- Column specification -----
## Delimiter: "\t"
## dbl (39): ID_REF, GSM1898106, GSM1898107, GSM1898108, GSM1898109, GSM1898110...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## File stored at:
##
## /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GPL4133.annot.gz

## Warning: One or more parsing issues, see `problems()` for details
if (length(gset) > 1) idx <- grep("GPL4133", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]

# make proper column names to match toptable
fvarLabels(gset) <- make.names(fvarLabels(gset))

# group membership for all samples
```

```

gsm <- "00000000000100000000000000000000100000000"
sm <- strsplit(gsm, split="")[[1]]

ex_org <- exprs(gset)
# quantile normalization
ex <- preprocessCore::normalize.quantiles(ex_org, copy = TRUE)
colnames(ex) <- colnames(ex_org)
rownames(ex) <- rownames(ex_org)

# log2 transformation
qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))
LogC <- (qx[5] > 100) ||
          (qx[6]-qx[1] > 50 && qx[2] > 0)
if (LogC) { ex[which(ex <= 0)] <- NaN
  exprs(gset) <- log2(ex) }

# assign samples to groups and set up design matrix
gs <- factor(sm)
groups <- make.names(c("day3","day4"))
levels(gs) <- groups
gset$group <- gs
design <- model.matrix(~group + 0, gset)
colnames(design) <- levels(gs)

fit <- lmFit(gset, design) # fit linear model

# set up contrasts of interest and recalculate model coefficients
cts <- paste(groups[1], groups[2], sep="-")
cont.matrix <- makeContrasts(contrasts=cts, levels=design)
fit2 <- contrasts.fit(fit, cont.matrix)

# compute statistics and table of top significant genes
fit2 <- eBayes(fit2, 0.01)

## Warning: Zero sample variances detected, have been offset away from zero
tT <- topTable(fit2, adjust="fdr", sort.by="B", number=250)

tT <- subset(tT, select=c("ID","adj.P.Val","P.Value","t","B","logFC","Gene.symbol","Gene.title"))
head(tT)

##           ID adj.P.Val     P.Value      t      B    logFC Gene.symbol
## 1173    1173 7.6261e-23 1.6941e-27 -29.3664 44.150 -1.47417      CH25H
## 14278   14278 1.6899e-15 7.5079e-20 -17.7543 32.205 -1.21599      DZIP1L
## 8433    8433 3.3126e-10 2.2077e-14 -11.9721 21.732 -1.99001      CCL8
## 42253   42253 2.6888e-09 2.3893e-13 -11.0415 19.625 -0.68293 KIAA1217
## 17700   17700 7.3215e-07 8.1323e-11 -8.9261 14.344 -0.68183 HNRNPLL
## 37962   37962 1.0804e-06 1.4401e-10 -8.7293 13.819 -0.91595      GALM
##                                     Gene.title
## 1173                      cholesterol 25-hydroxylase
## 14278      DAZ interacting zinc finger protein 1 like
## 8433                  C-C motif chemokine ligand 8
## 42253                           KIAA1217
## 17700 heterogeneous nuclear ribonucleoprotein L like
## 37962      galactose mutarotase

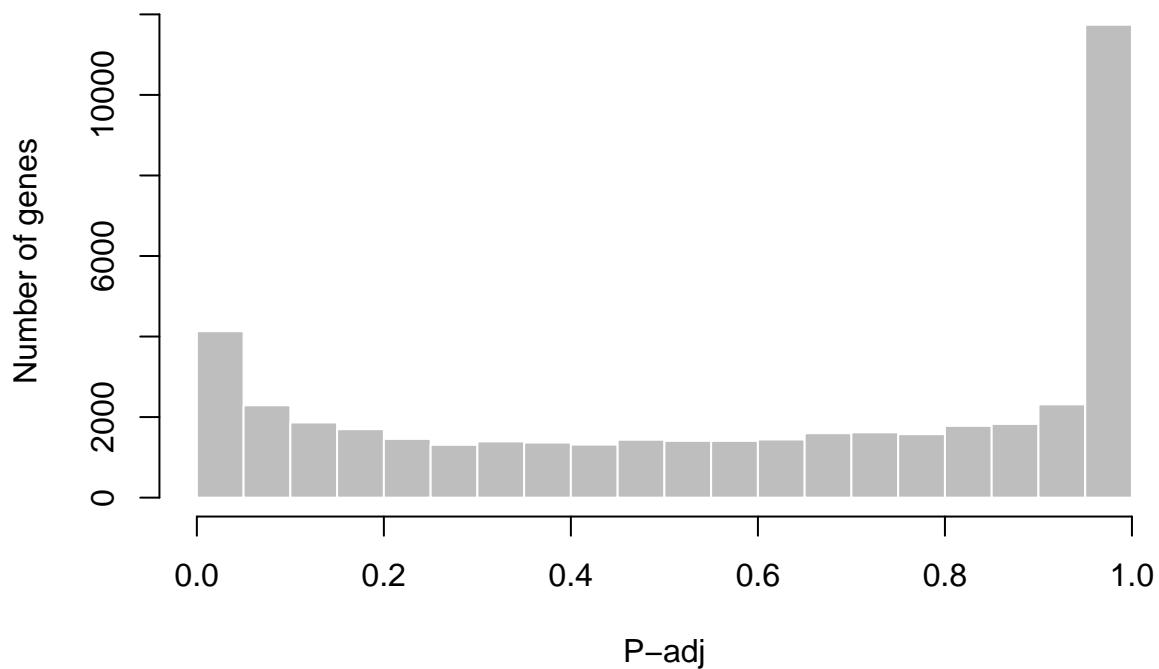
```

```

# Visualize and quality control test results.
# Build histogram of P-values for all genes. Normal test
# assumption is that most genes are not differentially expressed.
tT2 <- topTable(fit2, adjust="fdr", sort.by="B", number=Inf)
hist(tT2$adj.P.Val, col = "grey", border = "white", xlab = "P-adj",
     ylab = "Number of genes", main = "P-adj value distribution")

```

## P-adj value distribution

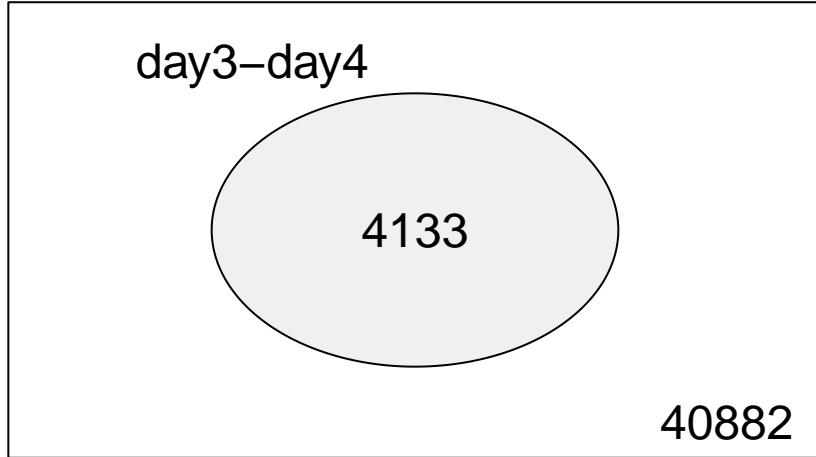


```

# summarize test results as "up", "down" or "not expressed"
dT <- decideTests(fit2, adjust.method="fdr", p.value=0.05)

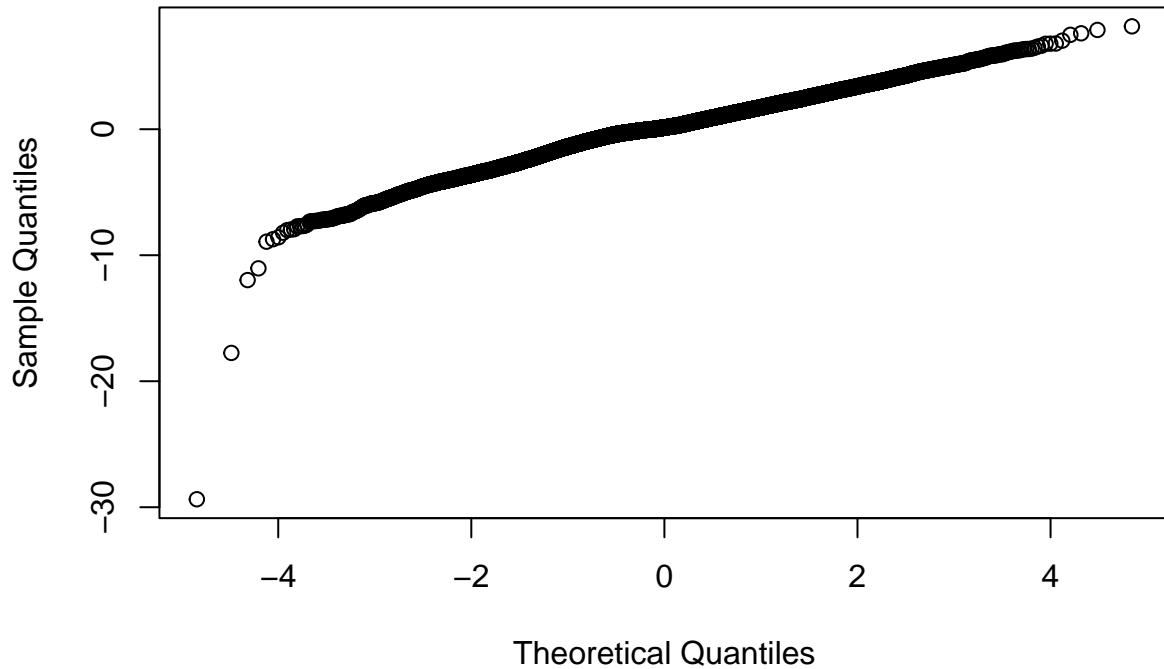
# Venn diagram of results
vennDiagram(dT, circle.col=palette())

```



```
# create Q-Q plot for t-statistic
t.good <- which(!is.na(fit2$F)) # filter out bad probes
qqt(fit2$t[t.good], fit2$df.total[t.good], main="Moderated t statistic")
```

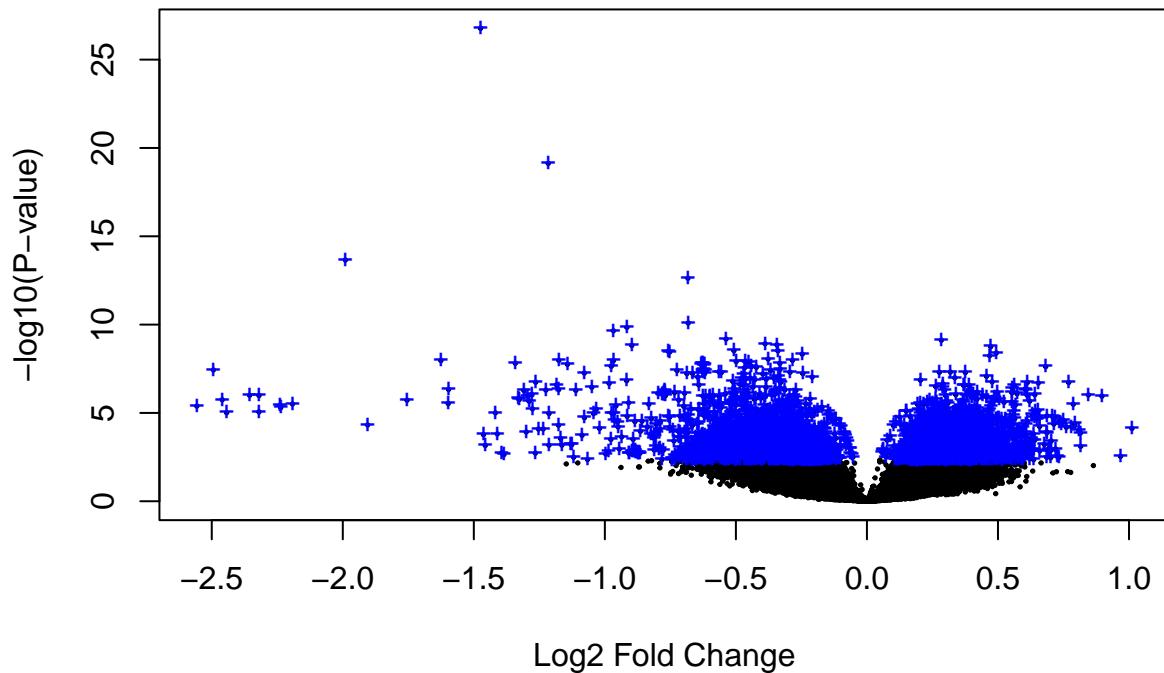
## Moderated t statistic



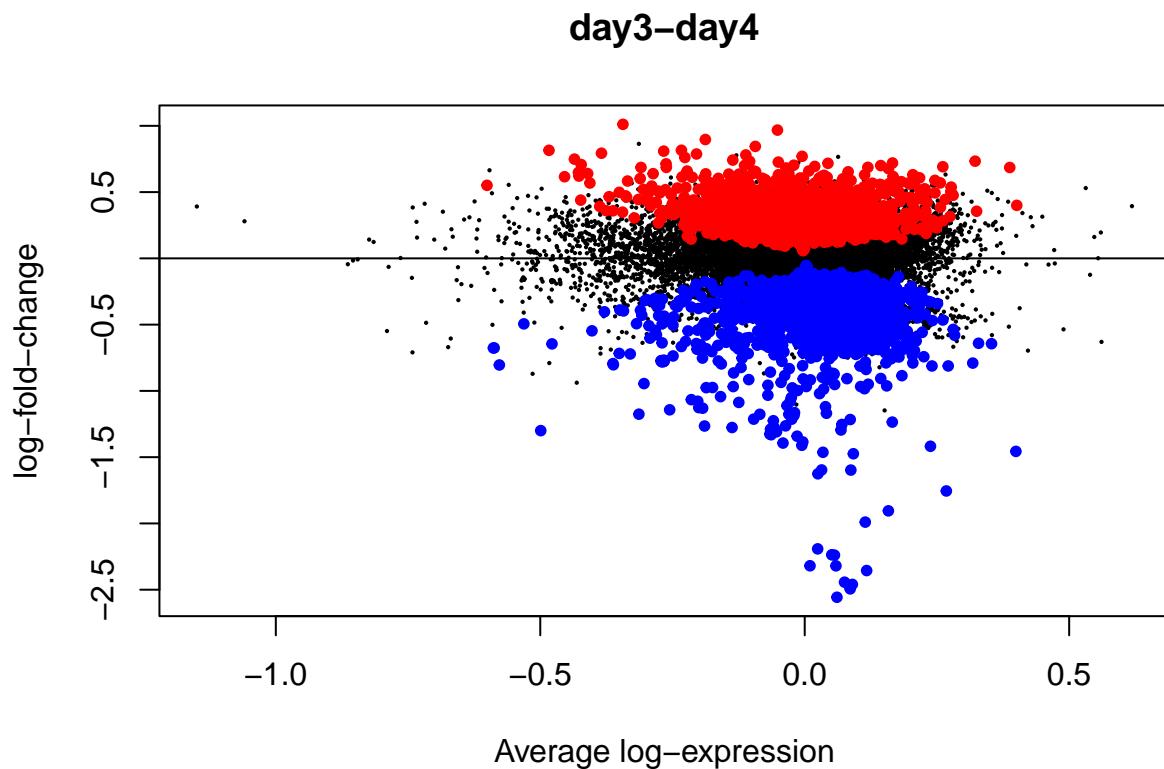
```
# volcano plot (log P-value vs log fold change)
colnames(fit2) # list contrast names

## [1] "day3-day4"
ct <- 1          # choose contrast of interest
volcanoplot(fit2, coef=ct, main=colnames(fit2)[ct], pch=20,
highlight=length(which(dT[,ct]!=0)), names=rep('+', nrow(fit2)))
```

day3–day4



```
# MD plot (log fold change vs mean log expression)
# highlight statistically significant (p-adj < 0.05) probes
plotMD(fit2, column=ct, status=dT[,ct], legend=F, pch=20, cex=1)
abline(h=0)
```



## RNaseq

Adapted from this vignette: (<https://bioconductor.org/packages/release/workflows/vignettes/RNAseq123/inst/doc/limmaWorkflows.html>)

### Data

To get started with this analysis, we'll download the raw file for GSE63310, and extract the relevant files from this archive. We'll use 9 samples.

```
url <- "https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE63310&format=file"
tmp_file <- file.path(tempdir(), "GSE63310_RAW.tar")
download.file(url, destfile = tmp_file, mode = "wb")
untar(tmp_file, exdir = dirname(tmp_file))

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

While each of the text files can be read into R separately and combined into a matrix of counts, `edgeR` offers a convenient way to do this in one step using the `readDGE()` function. The resulting `DGEList`-object contains a matrix of counts with 27179 rows associated with unique Entrez gene identifiers (IDs) and 9 columns associated with the individual samples in the experiment.

```

read.delim(files[1], nrow = 5)

##      EntrezID GeneLength Count
## 1      497097     3634     1
## 2    100503874     3259     0
## 3   100038431     1634     0
## 4     19888      9747     0
## 5    20671      3130     1

x <- readDGE(files, columns=c(1,3))
class(x)

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

## [1] 27179      9

```

## Organize Sample Information

```

samplenames <- substring(basename(colnames(x)), 12, nchar(basename(colnames(x))) - 4)
samplenames

## [1] "10_6_5_11" "9_6_5_11"  "purep53"    "JMS8-2"      "JMS8-3"      "JMS8-4"
## [7] "JMS8-5"     "JMS9-P7c"   "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
## 10_6_5_11 /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545535_10_6_5_11.txt.gz
## 9_6_5_11   /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545536_9_6_5_11.txt.gz
## purep53   /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545538_purep53.txt.gz
## JMS8-2    /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545539_JMS8-2.txt.gz
## JMS8-3    /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545540_JMS8-3.txt.gz
## JMS8-4    /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545541_JMS8-4.txt.gz
## JMS8-5    /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545542_JMS8-5.txt.gz
## JMS9-P7c  /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545544_JMS9-P7c.txt.gz
## JMS9-P8c  /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545545_JMS9-P8c.txt.gz
##           group lib.size norm.factors lane
## 10_6_5_11   LP 32863052          1 L004
## 9_6_5_11    ML 35335491          1 L004
## purep53    Basal 57160817        1 L004
## JMS8-2     Basal 51368625        1 L006
## JMS8-3     ML 75795034          1 L006
## JMS8-4     LP 60517657          1 L006
## JMS8-5     Basal 55086324        1 L006
## JMS9-P7c   ML 21311068          1 L008

```

```
## JMS9-P8c      LP 19958838
```

```
1 L008
```

## Organize Gene Annotations

A second data frame named genes in the DGEList-object is used to store gene-level information associated with rows of the counts matrix. This information can be retrieved using organism specific packages such as Mus.musculus for mouse (or Homo.sapiens for human) or the biomaRt package which interfaces the Ensembl genome databases in order to perform gene annotation.

The type of information that can be retrieved includes gene symbols, gene names, chromosome names and locations, Entrez gene IDs, Refseq gene IDs and Ensembl gene IDs to name just a few. biomaRt primarily works off Ensembl gene IDs, whereas Mus.musculus packages information from various sources and allows users to choose between many different gene IDs as the key.

The Entrez gene IDs available in our dataset were annotated using the Mus.musculus package to retrieve associated gene symbols and chromosome information.

```
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
# keeping only the first occurrence of each gene ID:
genes <- genes[!duplicated(genes$ENTREZID),]

x$genes <- genes
x

## An object of class "DGEList"
## $samples
##                               files
## 10_6_5_11 /var/folders/n0/hxwj6lwd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545535_10_6_5_11.txt.gz
## 9_6_5_11  /var/folders/n0/hxwj6lwd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545536_9_6_5_11.txt.gz
## purep53   /var/folders/n0/hxwj6lwd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545538_purep53.txt.gz
## JMS8-2    /var/folders/n0/hxwj6lwd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545539_JMS8-2.txt.gz
## JMS8-3    /var/folders/n0/hxwj6lwd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545540_JMS8-3.txt.gz
## JMS8-4    /var/folders/n0/hxwj6lwd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545541_JMS8-4.txt.gz
## JMS8-5    /var/folders/n0/hxwj6lwd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545542_JMS8-5.txt.gz
## JMS9-P7c  /var/folders/n0/hxwj6lwd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545544_JMS9-P7c.txt.gz
## JMS9-P8c  /var/folders/n0/hxwj6lwd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545545_JMS9-P8c.txt.gz
##           group lib.size norm.factors lane
## 10_6_5_11    LP 32863052        1 L004
## 9_6_5_11     ML 35335491        1 L004
## purep53     Basal 57160817        1 L004
## JMS8-2      Basal 51368625        1 L006
## JMS8-3      ML 75795034        1 L006
## JMS8-4      LP 60517657        1 L006
```

```

## JMS8-5      Basal 55086324          1 L006
## JMS9-P7c    ML 21311068           1 L008
## JMS9-P8c    LP 19958838           1 L008
##
## $counts
##             Samples
## Tags        10_6_5_11 9_6_5_11 purep53 JMS8-2 JMS8-3 JMS8-4 JMS8-5 JMS9-P7c
## 497097       1         2     342   526    3     3     535    2
## 100503874    0         0     5     6     0     0     5     0
## 100038431    0         0     0     0     0     0     1     0
## 19888        0         1     0     0     17    2     0     1
## 20671        1         1     76    40     33    14    98    18
##             Samples
## Tags        JMS9-P8c
## 497097       0
## 100503874    0
## 100038431    0
## 19888        0
## 20671        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 ...

```

## Data Pre-processing

For differential expression and related analyses, gene expression is rarely considered at the level of raw counts since libraries sequenced at a greater depth will result in higher counts. Rather, it is common practice to transform raw counts onto a scale that accounts for such library size differences. Popular transformations include counts per million (CPM), log2-counts per million (log-CPM), reads per kilobase of transcript per million (RPKM), and fragments per kilobase of transcript per million (FPKM).

Here raw counts are converted to CPM and log-CPM values using the cpm function in edgeR. RPKM values are just as easily calculated as CPM values using the rpkm function in edgeR if gene lengths are available.

```

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

```

```

##    10_6_5_11      9_6_5_11      purep53      JMS8-2
##  Min. :-4.507   Min. :-4.507   Min. :-4.5074   Min. :-4.5074
##  1st Qu.:-4.507 1st Qu.:-4.507 1st Qu.:-4.5074 1st Qu.:-4.5074
##  Median :-0.685 Median :-0.359 Median :-0.0951 Median :-0.0901
##  Mean   : 0.171 Mean   : 0.331 Mean   : 0.4356 Mean   : 0.4089
##  3rd Qu. : 4.291 3rd Qu. : 4.560 3rd Qu. : 4.6008 3rd Qu. : 4.5475
##  Max.   :14.763  Max.   :13.495  Max.   :12.9570  Max.   :12.8513
##             JMS8-3      JMS8-4      JMS8-5      JMS9-P7c
##  Min.   :-4.507   Min.   :-4.507   Min.   :-4.5074   Min.   :-4.507
##  1st Qu.:-4.507 1st Qu.:-4.507 1st Qu.:-4.5074 1st Qu.:-4.507

```

```

## Median :-0.428   Median :-0.406   Median :-0.0715   Median :-0.170
## Mean   : 0.323   Mean   : 0.253   Mean   : 0.4043   Mean   : 0.371
## 3rd Qu.: 4.577   3rd Qu.: 4.320   3rd Qu.: 4.4251   3rd Qu.: 4.603
## Max.   :12.958   Max.   :14.852   Max.   :13.1949   Max.   :12.941
##       JMS9-P8c
## Min.   :-4.507
## 1st Qu.:-4.507
## Median :-0.330
## Mean   : 0.275
## 3rd Qu.: 4.436
## Max.   :14.010

```

The log-CPM values will be used for exploratory plots. When `log=TRUE`, the `cpm` function adds an offset to the CPM values before converting to the log2-scale. By default, the offset is  $2/L$  where 2 is the “prior count” and L is the average library size in millions, so the log-CPM values are related to the CPM values by  $\log_2(CPM + 2/L)$ .

For this dataset, the average library size is about 45.5 million, so L approx. 45.5 and the minimum log-CPM value for each sample becomes  $\log_2(2/45.5) = -4.51$ . In other words, a count of zero for this data maps to a log-CPM value of -4.51 after adding the prior count or offset:

```

L <- mean(x$samples$lib.size) * 1e-6
M <- median(x$samples$lib.size) * 1e-6
c(L, M)

## [1] 45.489 51.369

```

## Removing Lowly Expressed Genes

```

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

##
## FALSE  TRUE
## 22026  5153

keep.exprs <- filterByExpr(x, group = group)
x <- x[keep.exprs, , keep.lib.sizes = FALSE]
dim(x)

## [1] 16624      9

```

By default, the function keeps genes with about 10 read counts or more in a minimum number of samples, where the number of samples is chosen according to the minimum group sample size.

```

lcpm.cutoff <- log2(10/M + 2/L)
library(RColorBrewer)
nsamples <- ncol(x)
col <- brewer.pal(nsamples, "Paired")
par(mfrow=c(1,2))

plot(density(lcpm[,1]), col=col[1], lwd=2, ylim=c(0,0.26), las=2, main="", xlab="")
title(main="A. Raw data", xlab="Log-cpm")
abline(v=lcpm.cutoff, lty=3)
for (i in 2:nsamples){
den <- density(lcpm[,i])
lines(den$x, den$y, col=col[i], lwd=2)
}

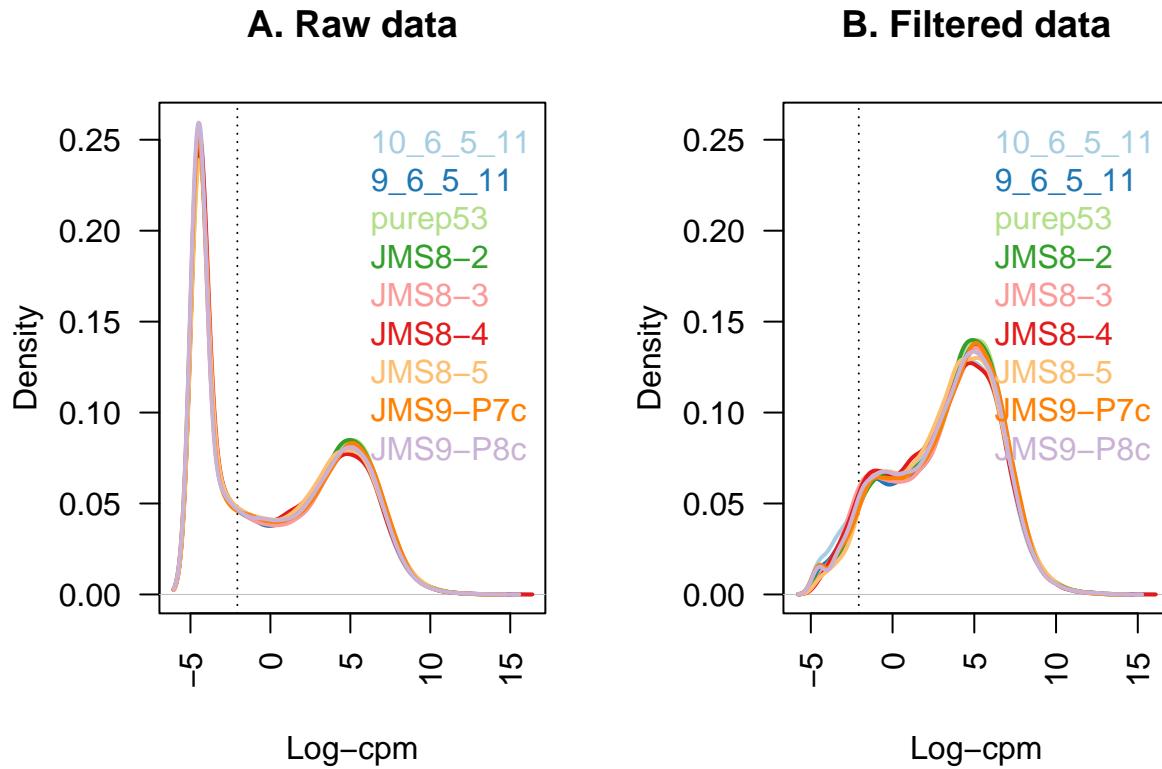
```

```

legend("topright", samplenames, text.col=col, bty="n")

lcpm <- cpm(x, log=TRUE)
plot(density(lcpm[,1]), col=col[1], lwd=2, ylim=c(0,0.26), las=2, main="", xlab="")
title(main="B. Filtered data", xlab="Log-cpm")
abline(v=lcpm.cutoff, lty=3)
for (i in 2:nsamples){
den <- density(lcpm[,i])
lines(den$x, den$y, col=col[i], lwd=2)
}
legend("topright", samplenames, text.col=col, bty="n")

```



## Normalization

Normalisation by the method of trimmed mean of M-values (TMM) is performed using the `calcNormFactors` function in `edgeR`. The normalization factors calculated here are used as a scaling factor for the library sizes. When working with `DGEList`-objects, these normalization factors are automatically stored in `x$samples$norm.factors`. For this dataset, the effect of TMM-normalisation is mild, as evident in the magnitude of the scaling factors, which are all relatively close to 1.

```

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

## [1] 0.89440 1.02502 1.04590 1.04585 1.01627 0.92171 0.99620 1.08610 0.98392

```

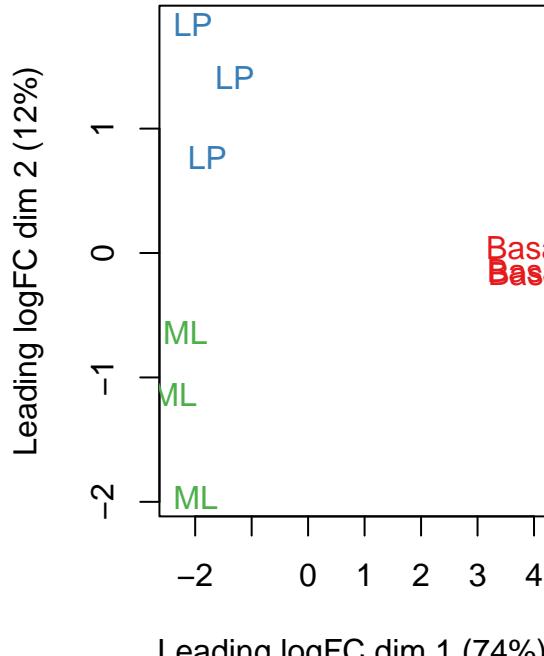
## Unsupervised clustering of samples

```

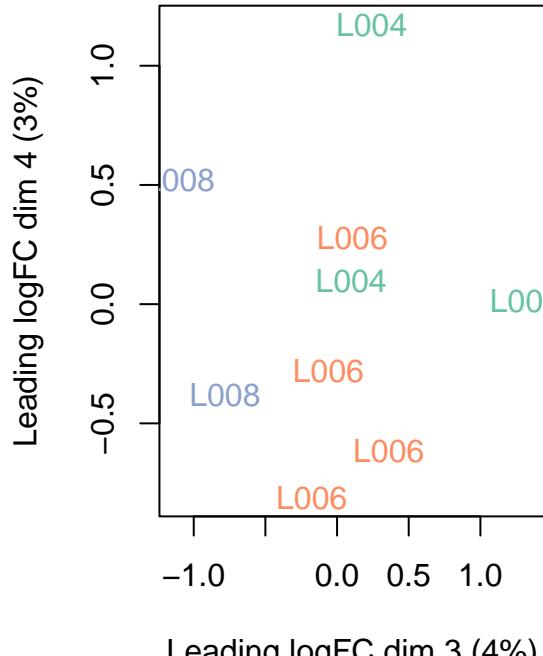
lcpm <- cpm(x, log=TRUE)
par(mfrow=c(1,2))
col.group <- group
levels(col.group) <- brewer.pal(nlevels(col.group), "Set1")
col.group <- as.character(col.group)
col.lane <- lane
levels(col.lane) <- brewer.pal(nlevels(col.lane), "Set2")
col.lane <- as.character(col.lane)
plotMDS(lcpm, labels=group, col=col.group)
title(main="A. Sample groups")
plotMDS(lcpm, labels=lane, col=col.lane, dim=c(3,4))
title(main="B. Sequencing lanes")

```

**A. Sample groups**



**B. Sequencing lanes**



## Differential Expression Analysis

### Creating a design matrix and contrasts

```

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"

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

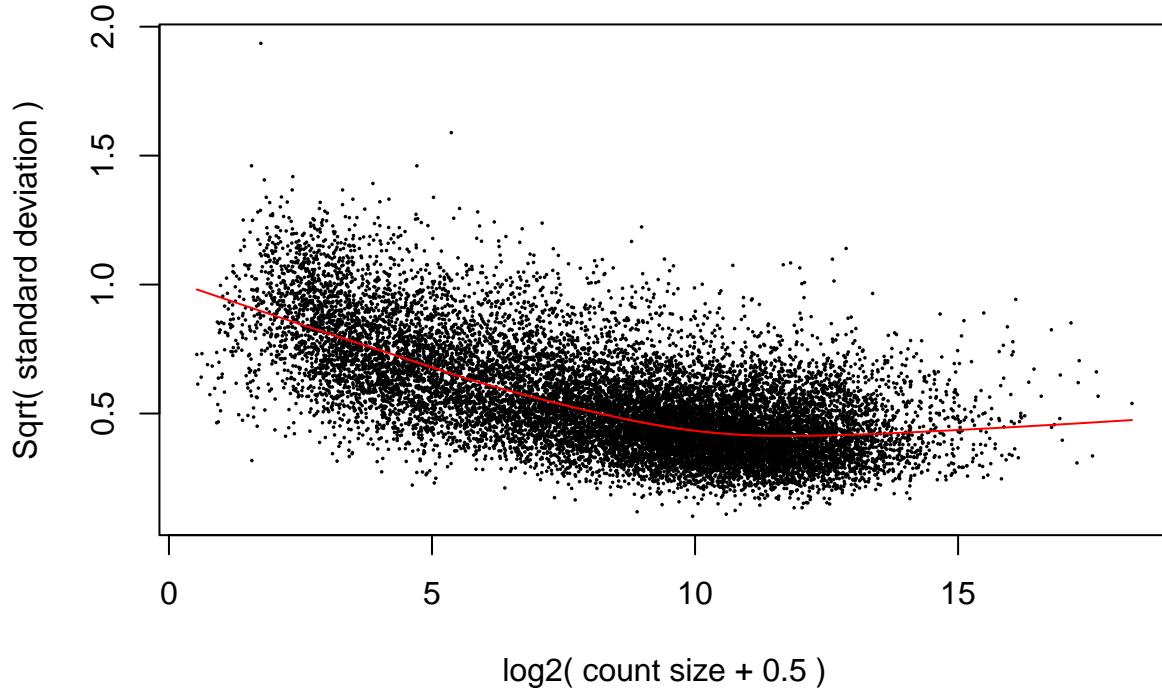
##          Contrasts
## Levels    BasalvsLP BasalvsML LPvsML
## Basal        1        1      0
## LP          -1        0      1
## ML           0       -1     -1
## laneL006      0        0      0
## laneL008      0        0      0

```

### Removing Heteroscedascity from Count Data

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

## voom: Mean-variance trend



v

```

## An object of class "EList"
## $genes
##   ENTREZID SYMBOL TXCHROM
## 1    497097  Xkr4    chr1
## 5    20671  Sox17    chr1
## 6    27395 Mrpl15    chr1
## 7    18777 Lypla1    chr1
## 9    21399 Tceal1    chr1
## 16619 more rows ...
##
## $targets
##                                         files
## 10_6_5_11 /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545535_10_6_5_11.txt.gz
## 9_6_5_11   /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545536_9_6_5_11.txt.gz
## purep53   /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545538_purep53.txt.gz
## JMS8-2    /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545539_JMS8-2.txt.gz
## JMS8-3    /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545540_JMS8-3.txt.gz
## JMS8-4    /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545541_JMS8-4.txt.gz
## JMS8-5    /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545542_JMS8-5.txt.gz
## JMS9-P7c  /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545544_JMS9-P7c.txt.gz
## JMS9-P8c  /var/folders/n0/hxwj61wd21s37_b12pzrdk480000gn/T//Rtmpe5JoGf/GSM1545545_JMS9-P8c.txt.gz
## group lib.size norm.factors lane
## 10_6_5_11    LP 29387429      0.89440 L004
## 9_6_5_11     ML 36212498      1.02502 L004
## purep53     Basal 59771061    1.04590 L004

```

```

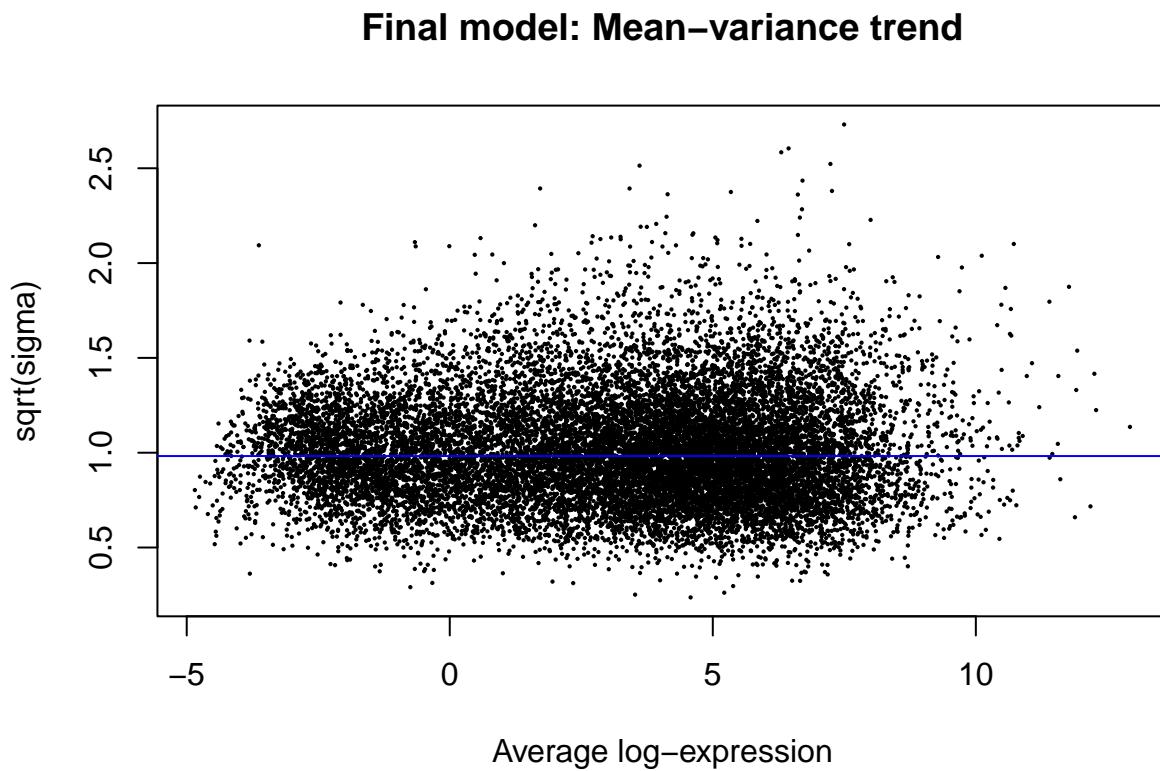
## JMS8-2    Basal 53711278      1.04585 L006
## JMS8-3      ML 77015912      1.01627 L006
## JMS8-4      LP 55769890      0.92171 L006
## JMS8-5    Basal 54863512      0.99620 L006
## JMS9-P7c    ML 23139691      1.08610 L008
## JMS9-P8c      LP 19634459      0.98392 L008
##
## $E
## Samples
## Tags      10_6_5_11 9_6_5_11 purep53  JMS8-2  JMS8-3  JMS8-4  JMS8-5 JMS9-P7c
## 497097    -4.2922  -3.8565 2.51858  3.2931 -4.4597 -3.9941 3.28697 -3.21037
## 20671     -4.2922  -4.5935 0.35601 -0.4073 -1.2010 -1.9434 0.84428 -0.32284
## 27395      3.8761   4.4131 4.51700  4.5618  4.3444  3.7864 3.89906  4.33961
## 18777      4.7088   5.5719 5.39640  5.1624  5.6494  5.0816 5.06025  5.75137
## 21399      4.7855   4.7545 5.37038  5.1221  4.8696  4.9438 5.13848  5.03090
## Samples
## Tags      JMS9-P8c
## 497097    -5.2953
## 20671     -1.2079
## 27395      4.1246
## 18777      5.1424
## 21399      4.9796
## 16619 more rows ...
##
## $weights
## [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]
## [1,] 1.0794 1.3330 19.8269 20.273 1.9937 1.3959 20.4950 1.1078 1.0794
## [2,] 1.1704 1.4564 4.8049 8.660 3.6125 2.6269 8.7601 3.2115 2.5419
## [3,] 20.2191 25.5738 30.4348 28.528 31.3523 25.7432 28.7225 21.2001 16.6579
## [4,] 26.9476 32.5059 33.5831 33.232 34.2318 32.3542 33.3343 30.3486 24.2598
## [5,] 26.6109 28.5016 33.6455 33.206 33.5735 31.9966 33.3085 25.1715 23.5733
## 16619 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"

```

```

vfit <- lmFit(v, design)
vfit <- contrasts.fit(vfit, contrasts = contr.matrix)
efit <- eBayes(vfit)
plotSA(efit, main="Final model: Mean-variance trend")

```



#### Examining the number of DEGs

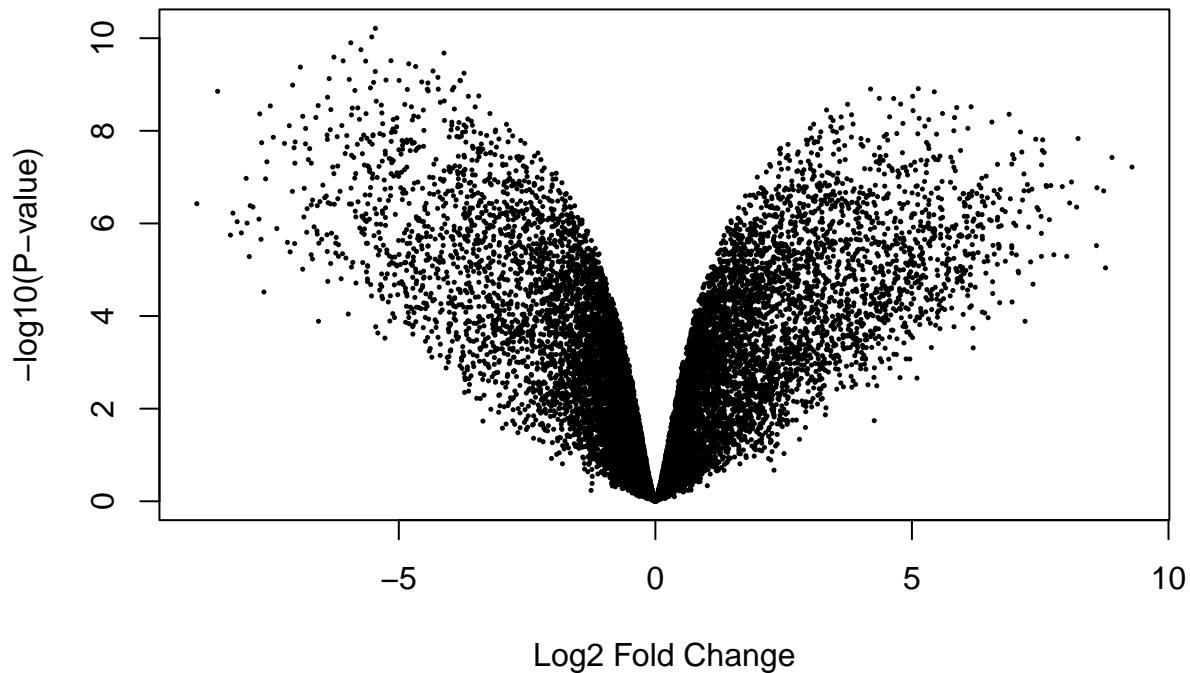
```

summary(decideTests(efit))

##          BasalvsLP BasalvsML LPvsML
## Down      4648     4927    3135
## NotSig    7113     7026   10972
## Up        4863     4671    2517

volcanoplot(efit, coef = 1)

```



```
basal.vs.lp <- topTreat(efit, coef = 1, n = Inf)
head(basal.vs.lp)
```

```
##      ENTREZID SYMBOL TXCHROM    logFC AveExpr       t   P.Value adj.P.Val
## 12759     12759   Clu    chr14 -5.4554  8.8566 -41.086 6.1264e-11 5.7747e-07
## 53624     53624  Cldn7    chr11 -5.5274  6.2954 -39.038 9.3746e-11 5.7747e-07
## 242505    242505  Rasef    chr4 -5.9352  5.1183 -37.683 1.2574e-10 5.7747e-07
## 67451     67451   Pkp2    chr16 -5.7387  4.4192 -36.157 1.7731e-10 5.7747e-07
## 12521     12521   Cd82    chr2 -4.1200  7.0696 -35.439 2.0947e-10 5.7747e-07
## 228543    228543   Rhov    chr2 -6.2642  5.4853 -34.598 2.5568e-10 5.7747e-07
##          B
## 12759  15.601
## 53624  15.119
## 242505 14.536
## 67451  14.250
## 12521  14.598
## 228543 14.169
library(gplots)

##
## Attaching package: 'gplots'

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

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

```
##  
##      space  
## The following object is masked from 'package:stats':  
##  
##      lowess  
  
basal.vs.lp.topgenes <- basal.vs.lp$ENTREZID[1:100]  
i <- which(v$genes$ENTREZID %in% basal.vs.lp.topgenes)  
mycol <- colorpanel(1000,"blue","white","red")  
# heatmap.2(lcpm[i,], scale="row",  
#   labRow=v$genes$SYMBOL[i], labCol=group,  
#   col=mycol, trace="none", density.info="none",  
#   margin=c(8,6), lhei=c(2,10), dendrogram="column")
```