2.RNA-Seq

Wanqing Tian

December 15, 2018

Load library

```
library(limma)
library(Glimma)
library(edgeR)
library(Mus.musculus)
library(R.utils)
library(RColorBrewer)
library(BiocManager)
library(RNAseq123)
library(RColorBrewer)
```

Combine Data

```
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(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
```

4 Data packaging

4.1 Reading in count-data

combine 9 chip data to 1 matrix by entrezID

```
x <- readDGE(files, columns=c(1,3))  # 1,3 entrezid, count
class(x)
```

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

```
dim(x)
```

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

```
DGEList(x)
```

```
## An object of class "DGEList"
## $counts
##            Samples
## Tags        GSM1545535_10_6_5_11 GSM1545536_9_6_5_11 GSM1545538_purep53
##   497097                       1                   2                342
##   100503874                    0                   0                  5
##   100038431                    0                   0                  0
##   19888                        0                   1                  0
##   20671                        1                   1                 76
##            Samples
## Tags        GSM1545539_JMS8-2 GSM1545540_JMS8-3 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        GSM1545542_JMS8-5 GSM1545544_JMS9-P7c 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 ...
##
## $samples
##                      group lib.size norm.factors
## GSM1545535_10_6_5_11     1 32863052            1
## GSM1545536_9_6_5_11      1 35335491            1
## GSM1545538_purep53       1 57160817            1
## GSM1545539_JMS8-2        1 51368625            1
## GSM1545540_JMS8-3        1 75795034            1
## GSM1545541_JMS8-4        1 60517657            1
## GSM1545542_JMS8-5        1 55086324            1
## GSM1545544_JMS9-P7c      1 21311068            1
## GSM1545545_JMS9-P8c      1 19958838            1
```

4.2 Organising sample information

Examples include cell type (basal, LP and ML in this experiment)

```
x$samples
```

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

```
## GSM1545545_JMS9-P8c   GSM1545545_JMS9-P8c.txt     1 19958838            1
```

```
samplenames <- substring(colnames(x), 12, nchar(colnames(x)))
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 group lib.size norm.factors lane
## 10_6_5_11 GSM1545535_10_6_5_11.txt    LP 32863052            1 L004
## 9_6_5_11   GSM1545536_9_6_5_11.txt    ML 35335491            1 L004
## purep53     GSM1545538_purep53.txt Basal 57160817            1 L004
## JMS8-2       GSM1545539_JMS8-2.txt Basal 51368625            1 L006
## JMS8-3       GSM1545540_JMS8-3.txt    ML 75795034            1 L006
## JMS8-4       GSM1545541_JMS8-4.txt    LP 60517657            1 L006
## JMS8-5       GSM1545542_JMS8-5.txt Basal 55086324            1 L006
## JMS9-P7c   GSM1545544_JMS9-P7c.txt    ML 21311068            1 L008
## JMS9-P8c   GSM1545545_JMS9-P8c.txt    LP 19958838            1 L008
```

4.3 Organising 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.

```
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
```

unique genes

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

add genes table in to data x

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

```
## An object of class "DGEList"
## $samples
##                             files group lib.size norm.factors lane
## 10_6_5_11 GSM1545535_10_6_5_11.txt    LP 32863052            1 L004
## 9_6_5_11   GSM1545536_9_6_5_11.txt    ML 35335491            1 L004
## purep53     GSM1545538_purep53.txt Basal 57160817            1 L004
## JMS8-2       GSM1545539_JMS8-2.txt Basal 51368625            1 L006
## JMS8-3       GSM1545540_JMS8-3.txt    ML 75795034            1 L006
## JMS8-4       GSM1545541_JMS8-4.txt    LP 60517657            1 L006
## JMS8-5       GSM1545542_JMS8-5.txt Basal 55086324            1 L006
## JMS9-P7c   GSM1545544_JMS9-P7c.txt    ML 21311068            1 L008
## JMS9-P8c   GSM1545545_JMS9-P8c.txt    LP 19958838            1 L008
##
## $counts
##            Samples
## Tags        10_6_5_11 9_6_5_11 purep53 JMS8-2 JMS8-3 JMS8-4 JMS8-5
##   497097            1        2     342    526      3      3    535
##   100503874         0        0       5      6      0      0      5
##   100038431         0        0       0      0      0      0      1
##   19888             0        1       0      0     17      2      0
##   20671             1        1      76     40     33     14     98
##            Samples
## Tags        JMS9-P7c JMS9-P8c
##   497097           2        0
##   100503874        0        0
##   100038431        0        0
##   19888            1        0
##   20671           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 ...
```

5 Data pre-processing

5.1 Transformations from the raw-scale

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

```
##            Samples
## Tags          10_6_5_11    9_6_5_11     purep53     JMS8-2      JMS8-3
##   497097     0.03042931  0.05660032  5.98311952 10.2397134  0.03958043
##   100503874  0.00000000  0.00000000  0.08747251  0.1168028  0.00000000
##   100038431  0.00000000  0.00000000  0.00000000  0.0000000  0.00000000
##   19888      0.00000000  0.02830016  0.00000000  0.0000000  0.22428910
##   20671      0.03042931  0.02830016  1.32958212  0.7786854  0.43538472
##   27395     13.11503265 21.81942229 23.93247808 24.6843282 20.63459725
##            Samples
## Tags             JMS8-4      JMS8-5    JMS9-P7c   JMS9-P8c
##   497097     0.04957231  9.71202943  0.09384795  0.0000000
##   100503874  0.00000000  0.09076663  0.00000000  0.0000000
##   100038431  0.00000000  0.01815333  0.00000000  0.0000000
##   19888      0.03304821  0.00000000  0.04692397  0.0000000
##   20671      0.23133744  1.77902595  0.84463153  0.4008249
##   27395     12.70703524 14.84942070 21.96041982 17.1352661
```

```
head(lcpm)
```

```
##            Samples
## Tags         10_6_5_11  9_6_5_11    purep53    JMS8-2    JMS8-3    JMS8-4
##   497097    -3.748623 -3.313765  2.5914607  3.362285 -3.581259 -3.418282
##   100503874 -4.507432 -4.507432 -2.9275280 -2.636931 -4.507432 -4.507432
##   100038431 -4.507432 -4.507432 -4.5074315 -4.507432 -4.507432 -4.507432
##   19888     -4.507432 -3.790514 -4.5074315 -4.507432 -1.898317 -3.698711
##   20671     -3.748623 -3.790514  0.4579085 -0.281645 -1.060843 -1.860900
##   27395      3.717978  4.450445  4.5835457  4.628091  4.370064  3.672539
##            Samples
## Tags            JMS8-5   JMS9-P7c  JMS9-P8c
##   497097     3.2862891 -2.8591947 -4.507432
##   100503874 -2.8918170 -4.5074315 -4.507432
##   100038431 -4.0087883 -4.5074315 -4.507432
##   19888     -4.5074315 -3.4597175 -4.507432
##   20671      0.8663089 -0.1703963 -1.168797
##   27395      3.8965999  4.4597191  4.102594
```

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

```
## [1] 45.48855 51.36862
```

```
summary(lcpm)
```

```
##    10_6_5_11          9_6_5_11          purep53             JMS8-2
##  Min.   :-4.5074   Min.   :-4.5074   Min.   :-4.50743   Min.   :-4.5074
##  1st Qu.:-4.5074   1st Qu.:-4.5074   1st Qu.:-4.50743   1st Qu.:-4.5074
##  Median :-0.6847   Median :-0.3589   Median :-0.09513   Median :-0.0901
```

```
##  Mean   : 0.1714   Mean   : 0.3312   Mean   : 0.43559   Mean   : 0.4089
##  3rd Qu.: 4.2913   3rd Qu.: 4.5601   3rd Qu.: 4.60081   3rd Qu.: 4.5475
##  Max.   :14.7632   Max.   :13.4952   Max.   :12.95700   Max.   :12.8513
##      JMS8-3            JMS8-4            JMS8-5            JMS9-P7c
##  Min.   :-4.5074   Min.   :-4.5074   Min.   :-4.50743   Min.   :-4.5074
##  1st Qu.:-4.5074   1st Qu.:-4.5074   1st Qu.:-4.50743   1st Qu.:-4.5074
##  Median :-0.4281   Median :-0.4064   Median :-0.07152   Median :-0.1704
##  Mean   : 0.3225   Mean   : 0.2529   Mean   : 0.40428   Mean   : 0.3708
##  3rd Qu.: 4.5772   3rd Qu.: 4.3199   3rd Qu.: 4.42513   3rd Qu.: 4.6031
##  Max.   :12.9578   Max.   :14.8520   Max.   :13.19491   Max.   :12.9413
##     JMS9-P8c
##  Min.   :-4.5074
##  1st Qu.:-4.5074
##  Median :-0.3300
##  Mean   : 0.2749
##  3rd Qu.: 4.4355
##  Max.   :14.0102
```

5.2 removing genes that are lowly expressed

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

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

Using a nominal CPM value of 1 (which is equivalent to a log-CPM value of 0)
The filterByExpr function in the edgeR package provides an automatic way to filter genes, while keeping as many genes as possible with worthwhile counts.
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. The actual filtering uses CPM values rather than counts in order to avoid giving preference to samples with large library sizes. For this dataset, the median library size is about 51 million and 10/51 approx. 0.2, so the filterByExpr function keeps genes that have a CPM of 0.2 or more in at least three samples. A biologically interesting gene should be expressed in at least three samples because all the cell type groups have three replicates. The cutoffs used depend on the sequencing depth and on the experimental design. If the library sizes had been larger then a lower CPM cutoff would have been chosen, because larger library sizes provide better resolution to explore more genes at lower expression levels. Alternatively, smaller library sizes decrease our ability to explore marginal genes and hence would have led to a higher CPM cutoff.

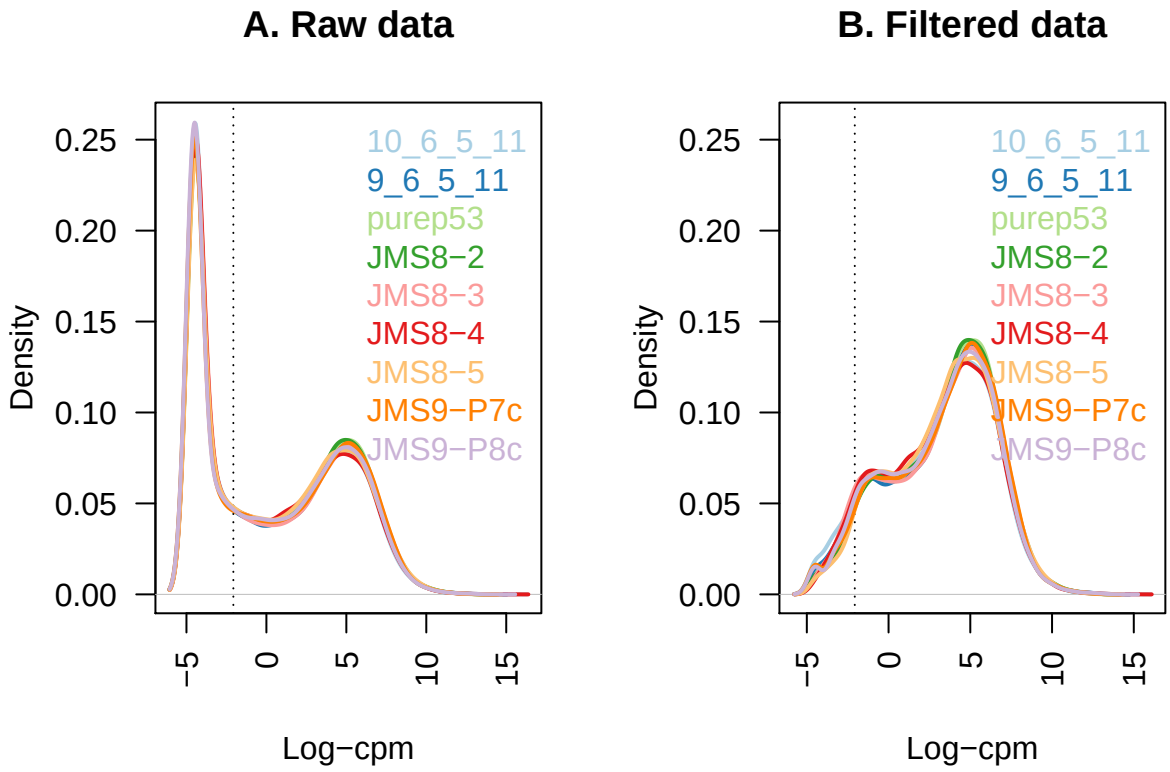
```
keep.exprs <- rowSums(cpm>1)>=3 #old
keep.exprs <- filterByExpr(x, group=group) # update

x <- x[keep.exprs,, keep.lib.sizes=FALSE]  #???
dim(x)
```

```
## [1] 16624     9
```

compare two graph, normal with reduce
cutoff line = Dotted vertical lines mark the log-CPM threshold (equivalent to a CPM value of about 0.2) used in the filtering step.

```
lcpm.cutoff <- log2(10/M + 2/L)   #Dotted vertical lines mark the log-CPM threshold (equivalent to a CP
```

```
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")
```



5.3 Normalising gene expression distributions

Nonetheless, normalisation by the method of trimmed mean of M-values (TMM) (Robinson and Oshlack 2010) is performed using the calcNormFactors function in edgeR.

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

```
## [1] 0.8943956 1.0250186 1.0459005 1.0458455 1.0162707 0.9217132 0.9961959
## [8] 1.0861026 0.9839203
```

```
x$samples
```

```
##                              files group lib.size norm.factors lane
## 10_6_5_11 GSM1545535_10_6_5_11.txt    LP 32857304    0.8943956 L004
## 9_6_5_11   GSM1545536_9_6_5_11.txt    ML 35328624    1.0250186 L004
## purep53     GSM1545538_purep53.txt Basal 57147943    1.0459005 L004
## JMS8-2       GSM1545539_JMS8-2.txt Basal 51356800    1.0458455 L006
## JMS8-3       GSM1545540_JMS8-3.txt    ML 75782871    1.0162707 L006
## JMS8-4       GSM1545541_JMS8-4.txt    LP 60506774    0.9217132 L006
## JMS8-5       GSM1545542_JMS8-5.txt Basal 55073018    0.9961959 L006
## JMS9-P7c   GSM1545544_JMS9-P7c.txt    ML 21305254    1.0861026 L008
## JMS9-P8c   GSM1545545_JMS9-P8c.txt    LP 19955335    0.9839203 L008
```

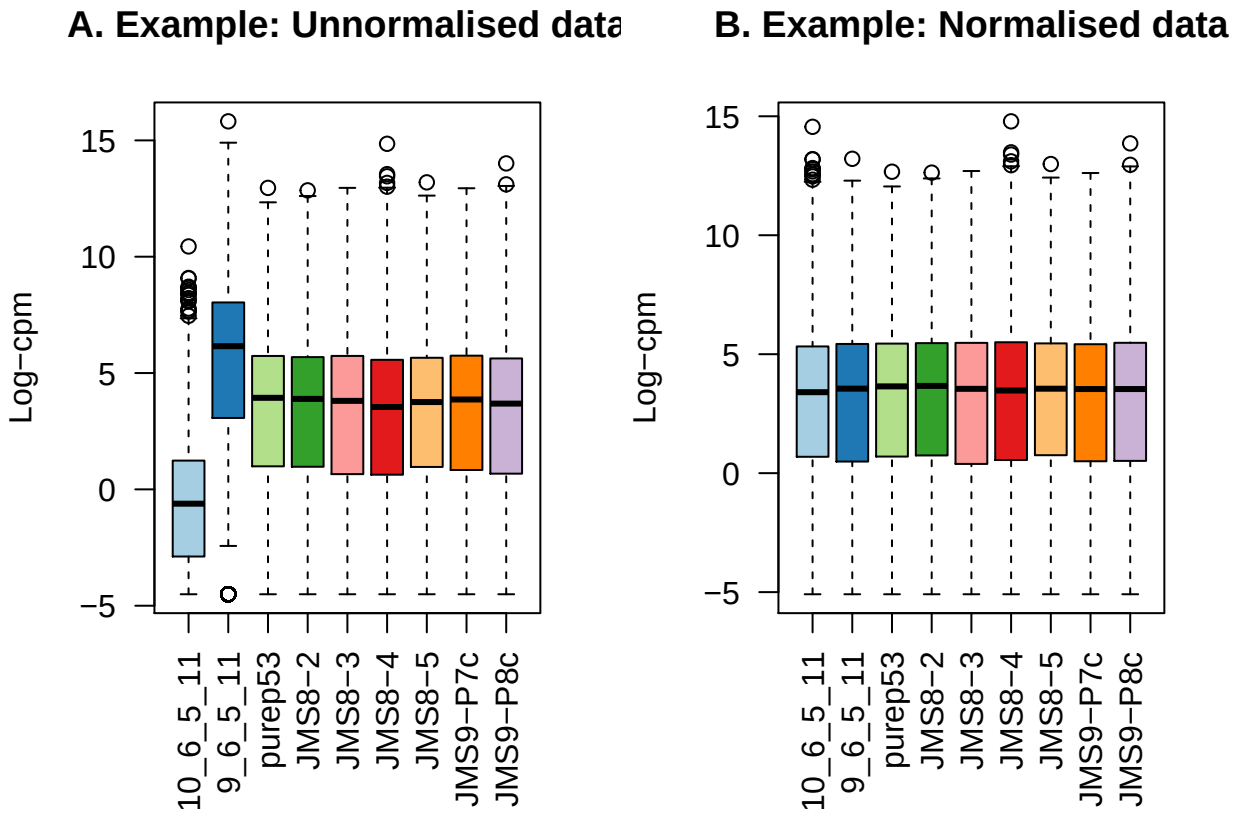
To give a better visual representation of the effects of normalisation, the data was duplicated then adjusted so that the counts of the first sample are reduced to 5% of their original values, and in the second sample they are inflated to be 5-times larger.

plot unnormalised data and normalised data

```
x2 <- x
x2$samples$norm.factors <- 1
x2$counts[,1] <- ceiling(x2$counts[,1]*0.05)
x2$counts[,2] <- x2$counts[,2]*5
par(mfrow=c(1,2))
lcpm <- cpm(x2, log=TRUE)
boxplot(lcpm, las=2, col=col, main="")
title(main="A. Example: Unnormalised data",ylab="Log-cpm")


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

```
## [1] 0.05770899 6.08287835 1.22023972 1.16478991 1.19661094 1.04659233
## [7] 1.15048074 1.25431164 1.10901983
```

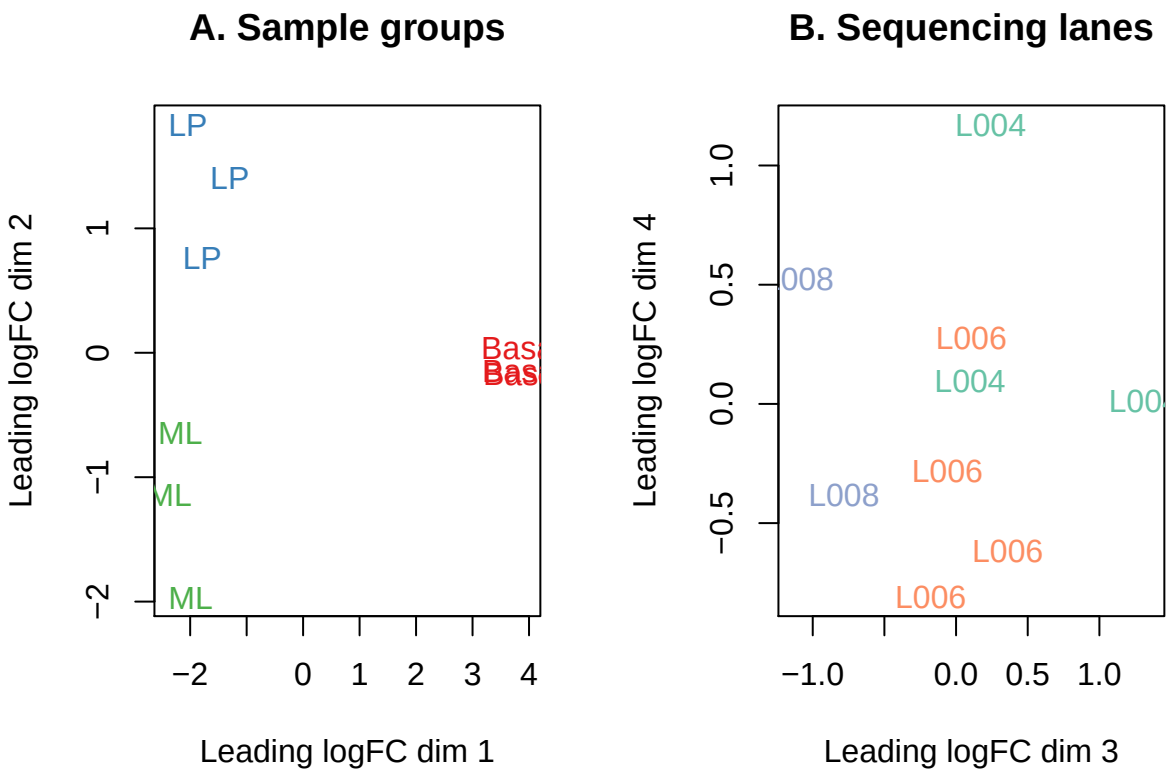
```
lcpm <- cpm(x2, log=TRUE)
boxplot(lcpm, las=2, col=col, main="")
title(main="B. Example: Normalised data",ylab="Log-cpm")
```



5.4 Unsupervised clustering of samples

brewer.pal ==set color

```
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")
```




```
glMDSPlot(lcpm, labels=paste(group, lane, sep="_"),
          groups=x$samples[,c(2,5)], launch=F)   ## change launch to T, there is a fancy graph open by l
```

6 Differential expression analysis

6.1 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"
```

For example, ~0+group+lane removes the intercept from the first factor, group, but an intercept remains in the second factor lane.
Contrasts for pairwise comparisons between cell populations are set up in limma using the makeContrasts function.

```
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
```

6.2 Removing heteroscedascity from count data

voom converts raw counts to log-CPM values by automatically extracting library sizes and normalisation factors from x itself.

```
par(mfrow=c(1,2))
v <- voom(x, design, plot=TRUE)
v   #plot left
```

```
## 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  Tcea1    chr1
## 16619 more rows ...
##
## $targets
##                              files group lib.size norm.factors lane
## 10_6_5_11 GSM1545535_10_6_5_11.txt    LP 29387429    0.8943956 L004
## 9_6_5_11   GSM1545536_9_6_5_11.txt    ML 36212498    1.0250186 L004
## purep53     GSM1545538_purep53.txt Basal 59771061    1.0459005 L004
## JMS8-2       GSM1545539_JMS8-2.txt Basal 53711278    1.0458455 L006
## JMS8-3       GSM1545540_JMS8-3.txt    ML 77015912    1.0162707 L006
## JMS8-4       GSM1545541_JMS8-4.txt    LP 55769890    0.9217132 L006
## JMS8-5       GSM1545542_JMS8-5.txt Basal 54863512    0.9961959 L006
## JMS9-P7c   GSM1545544_JMS9-P7c.txt    ML 23139691    1.0861026 L008
```

```
## JMS9-P8c   GSM1545545_JMS9-P8c.txt    LP 19634459    0.9839203 L008
##
## $E
##         Samples
## Tags     10_6_5_11  9_6_5_11   purep53     JMS8-2    JMS8-3    JMS8-4
##   497097 -4.292165 -3.856488 2.5185849  3.2931366 -4.459730 -3.994060
##   20671  -4.292165 -4.593453 0.3560126 -0.4073032 -1.200995 -1.943434
##   27395   3.876089  4.413107 4.5170045  4.5617546  4.344401  3.786363
##   18777   4.708774  5.571872 5.3964008  5.1623650  5.649355  5.081611
##   21399   4.785541  4.754537 5.3703795  5.1220551  4.869586  4.943840
##         Samples
## Tags        JMS8-5   JMS9-P7c  JMS9-P8c
##   497097 3.2869677 -3.2103696 -5.295316
##   20671  0.8442767 -0.3228444 -1.207853
##   27395  3.8990635  4.3396075  4.124644
##   18777  5.0602470  5.7513694  5.142436
##   21399  5.1384776  5.0308985  4.979644
## 16619 more rows ...
##
## $weights
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,]  1.079413  1.332986 19.826915 20.273253  1.993686  1.395853 20.494977
## [2,]  1.170357  1.456380  4.804866  8.660025  3.612508  2.626870  8.760149
## [3,] 20.219073 25.573792 30.434759 28.528310 31.352260 25.743247 28.722497
## [4,] 26.947557 32.505933 33.583128 33.232125 34.231754 32.354158 33.334340
## [5,] 26.610864 28.501638 33.645479 33.206374 33.573492 31.996623 33.308490
##           [,8]      [,9]
## [1,]  1.107780  1.079413
## [2,]  3.211473  2.541942
## [3,] 21.200072 16.657930
## [4,] 30.348630 24.259801
## [5,] 25.171513 23.573305
## 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")   #plot right
```

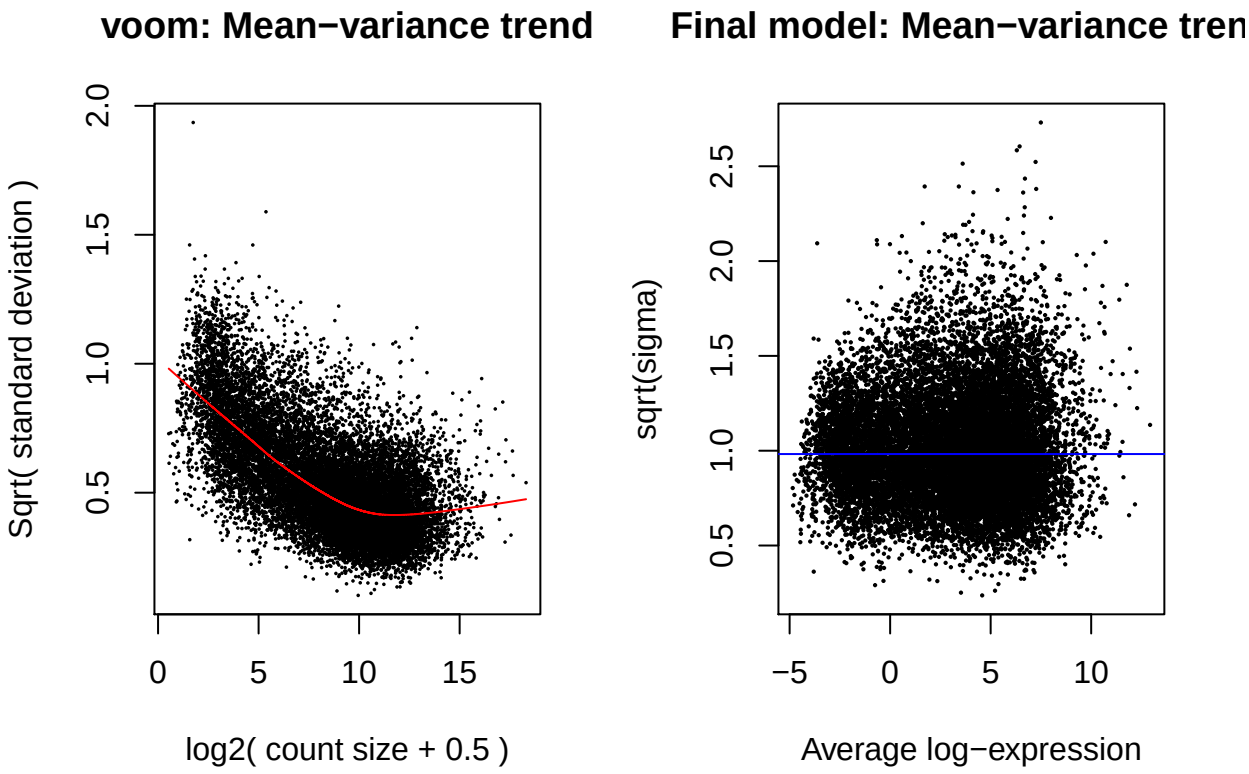


Figure 4: Means (x-axis) and variances (y-axis) of each gene are plotted to show the dependence between the two before voom is applied to the data (left panel) and how the trend is removed after voom precision weights are applied to the data (right panel)

6.3 Fitting linear models for comparisons of interest

Linear modelling in limma is carried out using the lmFit and contrasts.fit functions originally written for application to microarrays. The functions can be used for both microarray and RNA-seq data and fit a separate model to the expression values for each gene. Next, empirical Bayes moderation is carried out by borrowing information across all the genes to obtain more precise estimates of gene-wise variability (Smyth 2004). The model's residual variances are plotted against average expression values in the next figure. It can be seen from this plot that the variance is no longer dependent on the mean expression level.
##6.4 Examining the number of DE genes

For a quick look at differential expression levels, the number of significantly up- and down-regulated genes can be summarised in a table. Significance is defined using an adjusted p-value cutoff that is set at 5% by default.

```
summary(decideTests(efit))
```

```
##        BasalvsLP BasalvsML LPvsML
```

```
## Down        4648      4927   3135
## NotSig      7113      7026  10972
## Up          4863      4671   2517
```

For a stricter definition on significance, one may require log-fold-changes (log-FCs) to be above a minimum value. The treat method (McCarthy and Smyth 2009) can be used to calculate p-values from empirical Bayes moderated t-statistics with a minimum log-FC requirement.

```
tfit <- treat(vfit, lfc=1)   #Treat method adj log fold changes
dt <- decideTests(tfit)
summary(dt)
```

```
##        BasalvsLP BasalvsML LPvsML
## Down        1632      1777    224
## NotSig     12976     12790  16210
## Up          2016      2057    190
```

Genes that are DE in multiple comparisons can be extracted using the results from decideTests, where 0s represent genes that are not DE, 1s represent genes that are up-regulated, and -1s represent genes that are down-regulated.
The write.fit function can be used to extract and write results for all three comparisons to a single output file.

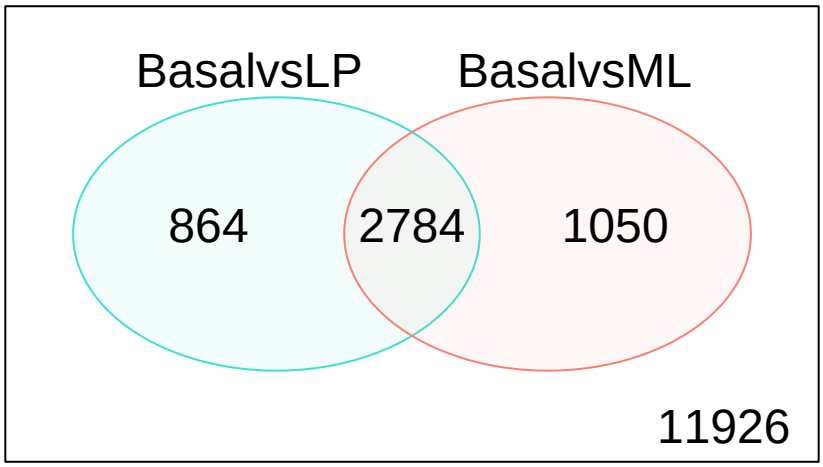
```
de.common <- which(dt[,1]!=0 & dt[,2]!=0)
length(de.common)
```

```
## [1] 2784
```

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

```
##  [1] "Xkr4"          "Rgs20"         "Cpa6"          "A830018L16Rik"
##  [5] "Sulf1"         "Eya1"          "Msc"           "Sbspon"
##  [9] "Pi15"          "Crispld1"      "Kcnq5"         "Rims1"
## [13] "Khdrbs2"       "Ptpn18"        "Prss39"        "Arhgef4"
## [17] "Cnga3"         "2010300C02Rik" "Aff3"          "Npas2"
```

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




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

6.5 Examining individual DE genes from top to bottom

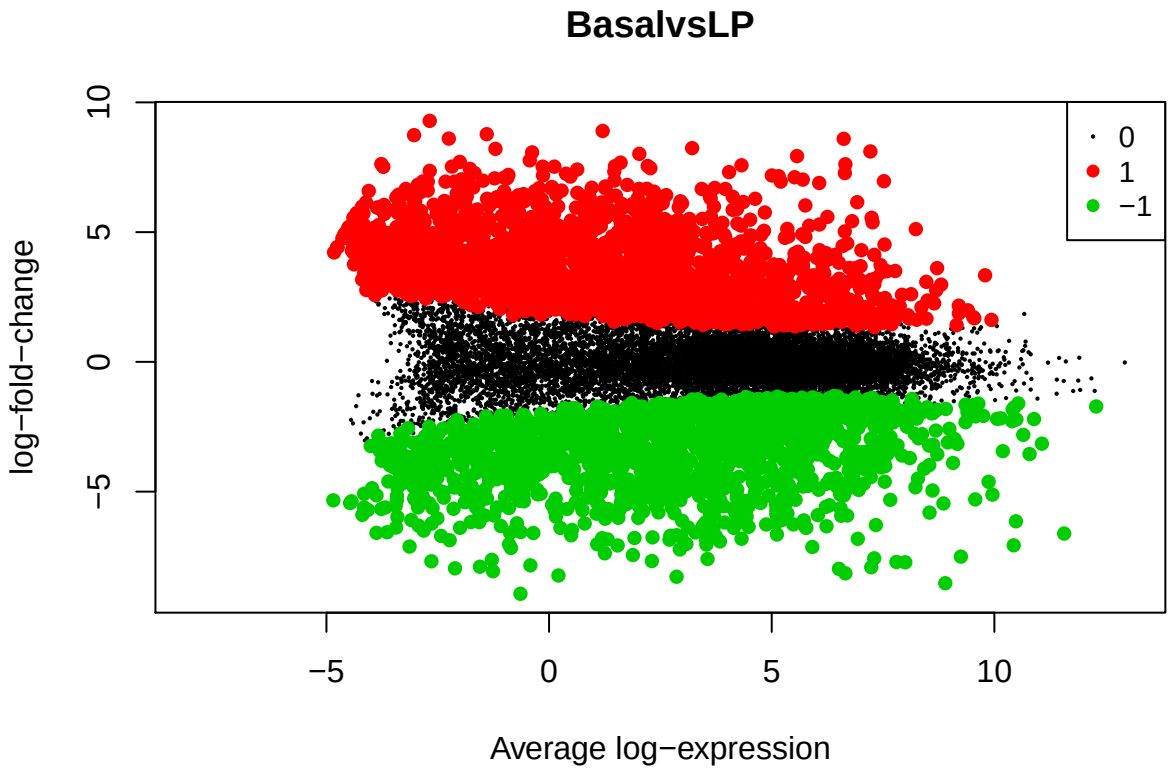
```
basal.vs.lp <- topTreat(tfit, coef=1, n=Inf)
basal.vs.ml <- topTreat(tfit, coef=2, n=Inf)
head(basal.vs.lp)
```

```
##        ENTREZID SYMBOL TXCHROM     logFC  AveExpr         t      P.Value
## 12759     12759    Clu   chr14 -5.455444 8.856581 -33.55508 1.723731e-10
## 53624     53624  Cldn7   chr11 -5.527356 6.295437 -31.97515 2.576972e-10
## 242505   242505  Rasef    chr4 -5.935249 5.118259 -31.33407 3.081544e-10
## 67451     67451   Pkp2   chr16 -5.738665 4.419170 -29.85616 4.575686e-10
## 228543   228543   Rhov    chr2 -6.264208 5.485314 -29.07484 5.782844e-10
## 70350     70350  Basp1   chr15 -6.084738 5.247023 -28.26649 7.267694e-10
##           adj.P.Val
## 12759  1.707586e-06
## 53624  1.707586e-06
## 242505 1.707586e-06
## 67451  1.739242e-06
## 228543 1.739242e-06
## 70350  1.739242e-06
```

6.6 Useful graphical representations of differential expression results

To summarise results for all genes visually, mean-difference plots, which display log-FCs from the linear model fit against the average log-CPM values can be generated using the plotMD function, with the differentially expressed genes highlighted.
Glimma extends this functionality by providing an interactive mean-difference plot via the glMDPlot function.

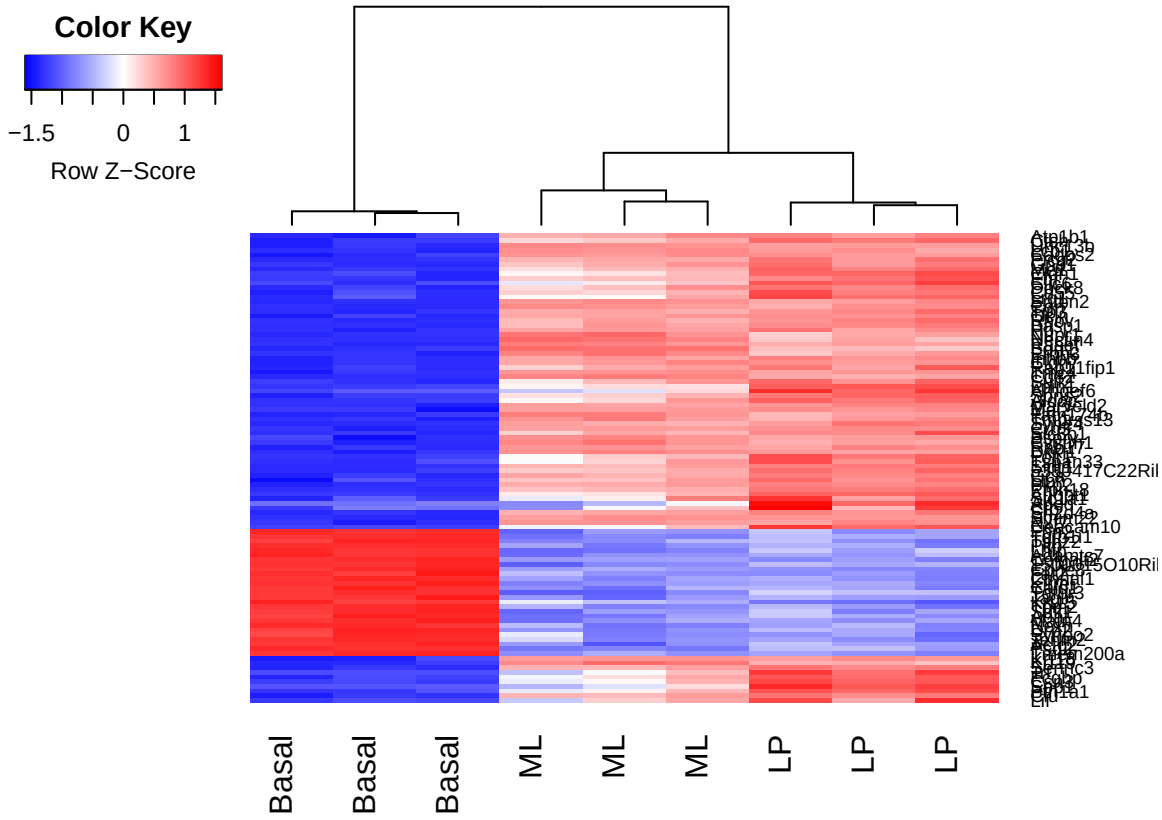
```
plotMD(tfit, column=1, status=dt[,1], main=colnames(tfit)[1],
       xlim=c(-8,13))
```




```
glMDPlot(tfit, coef=1, status=dt, main=colnames(tfit)[1],
         side.main="ENTREZID", counts=lcpm, groups=group, launch=F) ## glimma package
```

A heatmap is created for the top 100 DE genes (as ranked by adjusted p-value) from the basal versus LP contrast using the heatmap.2 function from the gplots package.

```
library(gplots)
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,key = TRUE,symkey = FALSE, trace="none", density.info="none", dendrogram="column")
```



7 Gene set testing with camera

```
load(system.file("extdata", "mouse_c2_v5p1.rda", package = "RNAseq123"))
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                        791        Up 1.768426e-18
## LIM_MAMMARY_STEM_CELL_DN                        683      Down 4.027711e-14
## ROSTY_CERVICAL_CANCER_PROLIFERATION_CLUSTER     170        Up 5.518131e-14
## LIM_MAMMARY_LUMINAL_PROGENITOR_UP                94      Down 2.737492e-13
## SOTIRIOU_BREAST_CANCER_GRADE_1_VS_3_UP          190        Up 5.155264e-13
##                                                    FDR
## LIM_MAMMARY_STEM_CELL_UP                     8.355813e-15
## LIM_MAMMARY_STEM_CELL_DN                     8.691057e-11
## ROSTY_CERVICAL_CANCER_PROLIFERATION_CLUSTER  8.691057e-11
## LIM_MAMMARY_LUMINAL_PROGENITOR_UP            3.233663e-10
## SOTIRIOU_BREAST_CANCER_GRADE_1_VS_3_UP       4.871724e-10
```

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

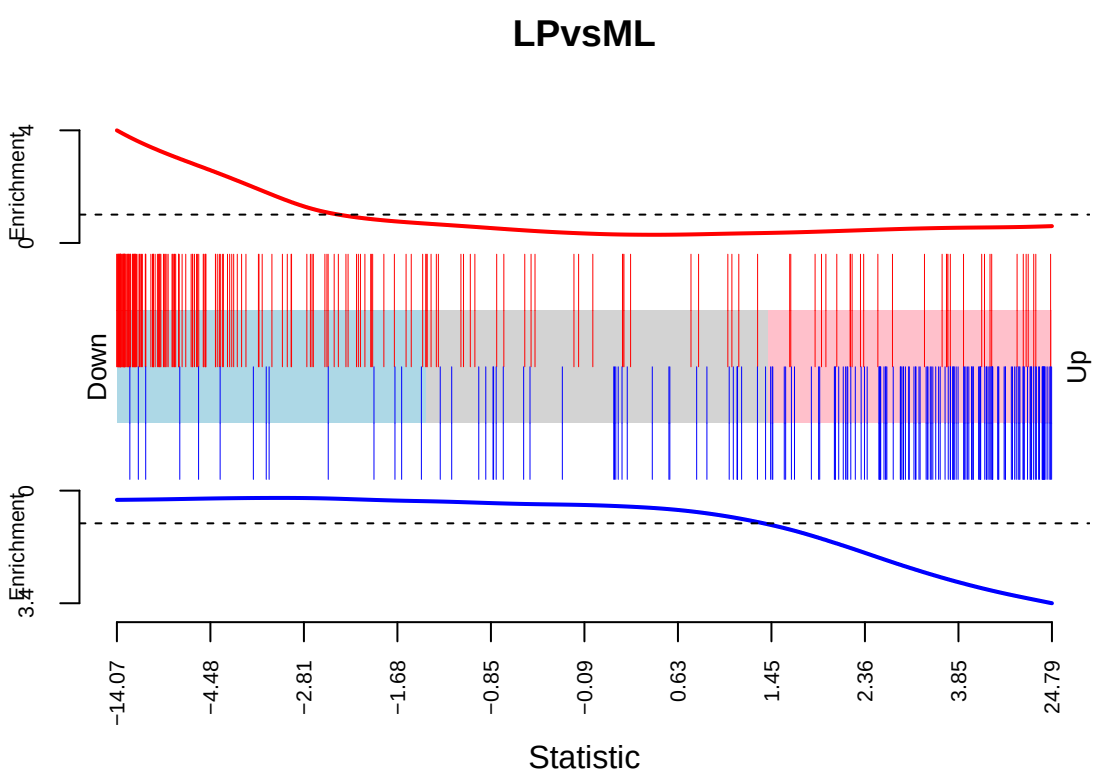
```
##                                        NGenes Direction       PValue
## LIM_MAMMARY_STEM_CELL_UP                  791        Up 1.676651e-22
## LIM_MAMMARY_STEM_CELL_DN                  683      Down 7.790412e-18
```

```
## LIM_MAMMARY_LUMINAL_MATURE_DN          172        Up 9.744096e-16
## LIM_MAMMARY_LUMINAL_MATURE_UP          204      Down 1.151832e-12
## NAKAYAMA_SOFT_TISSUE_TUMORS_PCA2_UP    137        Up 2.240253e-12
##                                             FDR
## LIM_MAMMARY_STEM_CELL_UP            7.922177e-19
## LIM_MAMMARY_STEM_CELL_DN            1.840485e-14
## LIM_MAMMARY_LUMINAL_MATURE_DN       1.534695e-12
## LIM_MAMMARY_LUMINAL_MATURE_UP       1.360602e-09
## NAKAYAMA_SOFT_TISSUE_TUMORS_PCA2_UP 1.880160e-09
```

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

```
##                                        NGenes Direction       PValue
## LIM_MAMMARY_LUMINAL_MATURE_DN             172        Up 6.733173e-14
## LIM_MAMMARY_LUMINAL_MATURE_UP             204      Down 9.967980e-14
## LIM_MAMMARY_LUMINAL_PROGENITOR_UP          94        Up 1.317692e-11
## REACTOME_RESPIRATORY_ELECTRON_TRANSPORT    94      Down 7.013195e-09
## REACTOME_RNA_POL_I_PROMOTER_OPENING        46      Down 2.037210e-08
##                                             FDR
## LIM_MAMMARY_LUMINAL_MATURE_DN          2.354935e-10
## LIM_MAMMARY_LUMINAL_MATURE_UP          2.354935e-10
## LIM_MAMMARY_LUMINAL_PROGENITOR_UP      2.075364e-08
## REACTOME_RESPIRATORY_ELECTRON_TRANSPORT 8.284337e-06
## REACTOME_RNA_POL_I_PROMOTER_OPENING    1.925163e-05
```

```
barcodeplot(efit$t[,3], index=idx$LIM_MAMMARY_LUMINAL_MATURE_UP,
            index2=idx$LIM_MAMMARY_LUMINAL_MATURE_DN, main="LPvsML")
```


LPvsML
Enrichment
4
0
Down
Up
Enrichment
0
−3.4
−14.07
−4.48
−2.81
−1.68
−0.85
−0.09
0.63
1.45
2.36
3.85
24.79
Statistic