Analysis of mammary glands

Kasper D. Hansen

Fall 2022

Data

Analysis is based on the paper “RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR” by Law et al (2018, F1000Res).

Data is RNA-seq data from the mammary glands of female mice in triplicate.

Cells have been sorted into 3 cell populations: basal, luminal progenitors (LP) and mature luminal (ML)

> *In this instance, reads were aligned to the mouse reference genome (mm10) using the R based pipeline available in the Rsubread package (specifically the align function followed by featureCounts for gene-level summarisation based on the in-built mm10 RefSeq-based annotation).*

Notes

Our focus here is on the steps performed in a typical RNA-seq analysis with voom. This is not a publication level analysis.

We only focus on 2 cell populations.

We don’t control for batch effects and we don’t model the fact that the cell populations are paired.

Loading the data

We load the `edgeR` package and the data in the form of a `RangedSummarizedExperiment` which might be familiar. `edgeR` uses its own data structure, a `DGEList`.

```r
library(edgeR)
load("mamgland.rda")
mamgland
dge <- SE2DGEList(mamgland[, mamgland$group %in% c("Basal", "LP")])
```

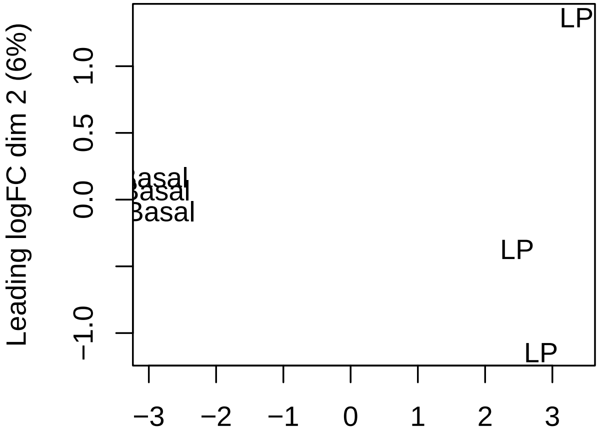
Inspecting the data

```
colData(mamgland)
head(rowData(mamgland))
```

There are many (27k) genes

MDS plot

```
plotMDS(dge, label =  dge$samples$group)
```



Expression filtering

```
keep.exprs <- filterByExpr(dge, group=dge$samples$group)
sum(keep.exprs)
```

```
## [1] 16253
```

```
dge <- dge[keep.exprs,, keep.lib.sizes=FALSE]
```

Size factor normalization

```r
dge <- calcNormFactors(dge, method = "TMM")
```

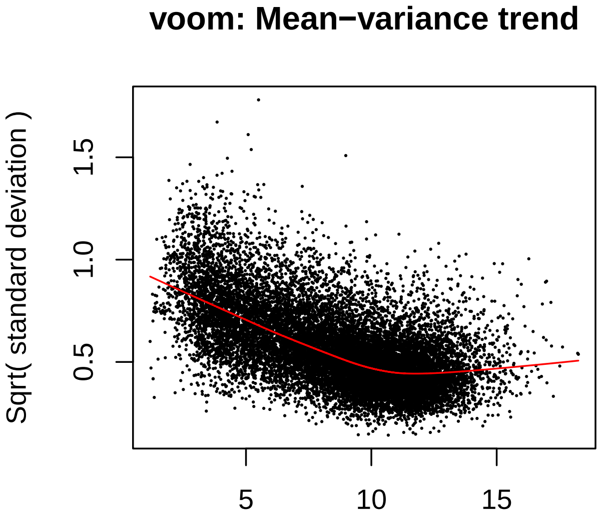
Design

```r
design <- model.matrix(~ group, data = dge$samples)
colnames(design) <- gsub("group", "", colnames(design))
design
```

```
##           (Intercept) LP
## 10_6_5_11           1  1
## purep53             1  0
## JMS8-2              1  0
## JMS8-4              1  1
## JMS8-5              1  0
## JMS9-P8c            1  1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
```

Mean-variance relationship

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

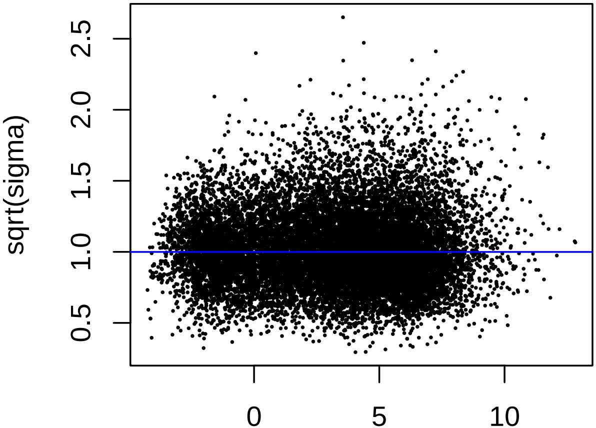


Fitting the model and variance shrinkage

```
vfit <- lmFit(v, design)
vfit <- eBayes(vfit)
```

Checking that the variance is stabilized

```
plotSA(vfit)
```



Inspecting the top genes

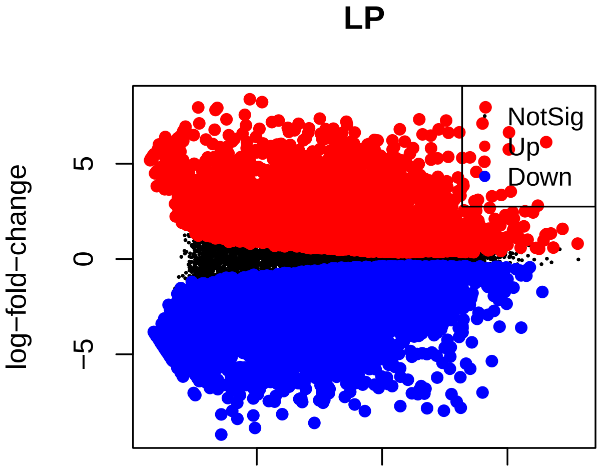
```r
topTable(vfit, n = 5)
```

```
## Removing intercept from test coefficients

##          rownames ENTREZID   SYMBOL TXCHROM     logFC  AveExpr         t
## 12994       18093    12994     Csn3    chr5  7.109058 9.004706  40.90067
## 20750       18275    20750     Spp1    chr5  7.959573 9.125674  34.24679
## 110935      19839   110935 Atp6v1b1    chr6  6.539368 6.613061  33.99703
## 269328      13648   269328    Muc15    chr2  4.985358 6.461080  33.24706
## 16664        4122    16664    Krt14   chr11 -7.804574 8.137070 -33.70876
##             P.Value    adj.P.Val        B
## 12994  5.168915e-12 8.401038e-08 17.71763
## 20750  2.768651e-11 1.100207e-07 16.33496
## 110935 2.966737e-11 1.100207e-07 16.23428
## 269328 3.662002e-11 1.100207e-07 16.15654
## 16664  3.215058e-11 1.100207e-07 16.06298
```

Mean-difference plot

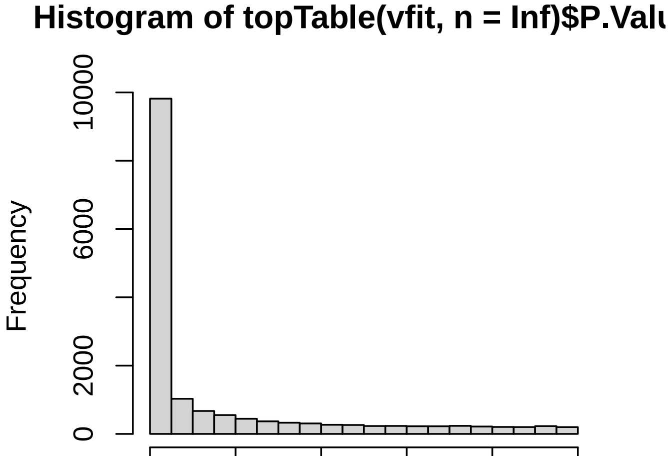
```
dt <-  decideTests(vfit)
plotMD(vfit, status = dt[,2])
```



Do thee p-values look good?

```
hist(topTable(vfit, n = Inf)$P.Value)
```

```
## Removing intercept from test coefficients
```


Histogram of topTable(vfit, n = Inf)$P.Valu
Frequency