# Analysis of mammary glands

## Using DESeq2

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).* — *from Law et at*

# Notes

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

We only focus on 2 out of the 3 cell populations, to make things simple.

We don't control for batch effects and we don't model the fact that the cell populations are paired, in the sense that one sample has been sorted into three different populations.

# 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.

```
library(DESeq2)
```

```
## Warning: package 'DESeq2' was built under R version 4.3.1
## Loading required package: S4Vectors
## Warning: package 'S4Vectors' was built under R version 4.3.1
## Loading required package: stats4
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
```

# Inspecting the data

```
mamgland_rse
```

```
## class: RangedSummarizedExperiment
## dim: 27179 9
## metadata(1): version
## assays(1): counts
## rownames(27179): 497097 100503874 ... 100861691 100504472
## rowData names(4): rownames ENTREZID SYMBOL TXCHROM
## colnames(9): 10_6_5_11 9_6_5_11 ... JMS9-P7c JMS9-P8c
## colData names(3): files group lane
```

```
colData(mamgland_rse)
```

```
## DataFrame with 9 rows and 3 columns
##                            files     group      lane
##                      <character> <factor> <factor>
## 10_6_5_11 GSM1545535_10_6_5_11    LP         L004
## 9_6_5_11   GSM1545536_9_6_5_11    ML         L004
## purep53     GSM1545538_purep53    Basal      L004
## JMS8-2       GSM1545539_JMS8-2    Basal      L006
## JMS8-3       GSM1545540_JMS8-3    ML         L006
## JMS8-4       GSM1545541_JMS8-4    LP         L006
## JMS8-5       GSM1545542_JMS8-5    Basal      L006
## JMS9-P7c   GSM1545544_JMS9-P7c    ML         L008
## JMS9-P8c   GSM1545545_JMS9-P8c    LP         L008
```

```
rowData(mamgland_rse)
```

```
## DataFrame with 27179 rows and 4 columns
##            rownames    ENTREZID       SYMBOL     TXCHROM
##           <integer> <character>  <character> <character>
## 497097            1      497097         Xkr4        chr1
## 100503874         2   100503874      Gm19938          NA
## 100038431         3   100038431      Gm10568          NA
## 19888             4       19888          Rp1        chr1
## 20671             5       20671        Sox17        chr1
## ...             ...         ...          ...         ...
## 100861837     27204   100861837           NA          NA
## 100861924     27205   100861924           NA          NA
## 170942        27206      170942       Erdr1x        chrY
## 100861691     27207   100861691 LOC100861691          NA
## 100504472     27208   100504472           NA          NA
```

There are many (27k) genes

# Making a DDS object

```
rse <- mamgland_rse[, mamgland_rse$group %in% c("Basal", "LP")]
dds <- DESeqDataSet(rse, design = ~ group)
```

```
## factor levels were dropped which had no samples
```

```
dds$group
```

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

We see that the reference level is `Basal`

# NOTES

1. MDS/PCA plot
2. Expression filtering
3. Setting up the design matrix
4. Size factor normalization
5. Mean-variance relationship
6. Interpreting the output
7. Fold change shrinking
8. Does the p-value distribution look good

# Fitting the model and variance shrinkage

First, we fit the model and extract results for a specific comparison of interest. In this case, it is all governed by

```
design = ~ group
```

argument to `DESeqData`. And because the model only contains a single comparison of interest, it is easy to get. We can use `resultNames()` to get a list of possibly interesting comparisons in the object.

```
dds <- DESeq(dds)
```

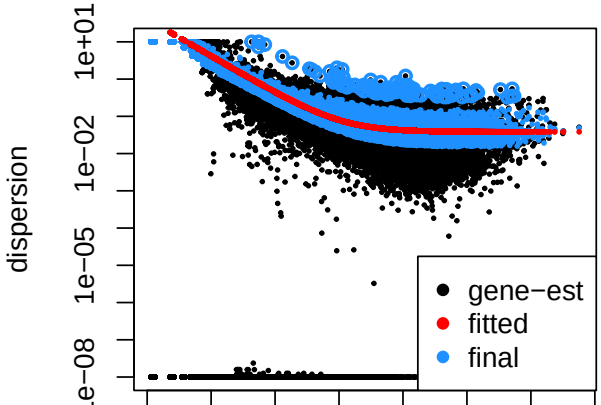
```
## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
```

```
resultsNames(dds)
```

# Checking that the variance is stabilized

DESeq2 makes it easy to look at the dispersion shrinkage happening "inside" of DESeq2.

```
plotDispEsts(dds)
```



# Inspecting the top genes

The output from `results()` has the same ordering as the data. Usually, we want to look

# Do the p-values look good?

# Getting better fold-change estimate by shrinkage

```
resultsNames(dds)
```

```
## [1] "Intercept"          "group_LP_vs_Basal"
```

```
resLFC <- lfcShrink(dds, coef = "group_LP_vs_Basal")
```

```
## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions
##     sequence count data: removing the noise and preserving large differences.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
```

```
plotMA(res)
```