# QLSC_Assignment_3.1

*Niklas Brake*

*October 22, 2018*

```r
library(DESeq2)
```

```
## Warning: package 'DESeq2' was built under R version 3.3.2
```

```
## Loading required package: S4Vectors
```

```
## Warning: package 'S4Vectors' was built under R version 3.3.3
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
## Warning: package 'BiocGenerics' was built under R version 3.3.1
```

```
## Loading required package: parallel
```

```
## 
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
## 
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

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

```
## The following objects are masked from 'package:base':
## 
##     anyDuplicated, append, as.data.frame, cbind, colnames,
##     do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, lengths, 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, which.max,
##     which.min
```

```
## 
## Attaching package: 'S4Vectors'
```

```
## The following objects are masked from 'package:base':
## 
##     colMeans, colSums, expand.grid, rowMeans, rowSums
```

```
## Loading required package: IRanges
```

```
## Warning: package 'IRanges' was built under R version 3.3.3
```

```
## Loading required package: GenomicRanges
```

```
## Warning: package 'GenomicRanges' was built under R version 3.3.3
```

```
## Loading required package: GenomeInfoDb
## Warning: package 'GenomeInfoDb' was built under R version 3.3.2
```

```
## Loading required package: SummarizedExperiment
```

```
## Warning: package 'SummarizedExperiment' was built under R version 3.3.1
```

```
## Loading required package: Biobase
```

```
## Warning: package 'Biobase' was built under R version 3.3.1
```

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

```r
count_matrix = read.table("C:\\Users\\brake\\Documents\\QLSC600\\Module 3\\QLS_counts.tsv");
sample_annotation = read.table("C:\\Users\\brake\\Documents\\QLSC600\\Module 3\\QLS_annotations.tsv");

dds = DESeqDataSetFromMatrix(countData = count_matrix, colData = sample_annotation, design = ~ group)
ddsT <-rlog(dds)
PCs = plotPCA(ddsT, intgroup = "group", ntop = 1000, returnData = TRUE)
PCs
```

```
##           PC1        PC2 group group.1 name
## A1 -23.363647 -15.345094     A       A   A1
## B1   8.169687 -17.097986     B       B   B1
## A2  10.806618   5.912861     A       A   A2
## A3 -17.018750  11.838609     A       A   A3
## B2  12.056107   6.688231     B       B   B2
## A4 -19.376134   8.334029     A       A   A4
## B3  13.227093  -2.077580     B       B   B3
## B4  15.499026   1.746930     B       B   B4
```

```r
dds = DESeq(dds)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```r
DE_Results = results(dds)
DE_Results[DE_Results$padj < 0.01 & !is.na(DE_Results$padj),]
```

```
## log2 fold change (MAP): group B vs A
## Wald test p-value: group B vs A
## DataFrame with 373 rows and 6 columns
##          baseMean log2FoldChange     lfcSE      stat       pvalue
##         <numeric>      <numeric> <numeric> <numeric>    <numeric>
## Sgk3    407.49165      0.8372438 0.2173943  3.851269 1.175074e-04
## Cpa6     45.22598      1.6037797 0.3296475  4.865136 1.143784e-06
## Prex2   655.67757      2.3947175 0.3181370  7.527316 5.179381e-14
## Rdh10  1092.63130     -1.3413982 0.2605671 -5.147996 2.632846e-07
## Il1rl1  635.15071     -1.8942450 0.3189523 -5.938960 2.868360e-09
## ...           ...           ...       ...       ...          ...
## Pcgf5    845.0844     -1.2136914 0.2626456 -4.621023 3.818516e-06
## Gsto1   5830.7654     -0.8061702 0.2036140 -3.959307 7.516747e-05
## Gsto2    169.3438     -1.4488935 0.3210616 -4.512821 6.397113e-06
## Add3    3837.7707      0.5792550 0.1332371  4.347550 1.376665e-05
## Dusp5    365.9472     -0.7429779 0.1740234 -4.269414 1.959871e-05
##                 padj
##            <numeric>
## Sgk3   5.582008e-03
## Cpa6   2.006168e-04
## Prex2  1.180985e-10
## Rdh10  6.210339e-05
## Il1rl1 1.401501e-06
## ...              ...
## Pcgf5  0.0004427214
## Gsto1  0.0041634256
## Gsto2  0.0006711232
## Add3   0.0012390887
## Dusp5  0.0016055687
```

One observes that PC1 segregates the data into group A and group B, with the exception of one A sample, who's PC1 is positive. This is sample A2.

```
count_matrix2 = count_matrix[,c(1:2,4:8)]
sample_annotation2 = sample_annotation[c(1:2,4:8),]
dds2 = DESeqDataSetFromMatrix(countData = count_matrix2, colData = sample_annotation2, design = ~ group)
dds2T <-rlog(dds2)
PCs2 = plotPCA(dds2T, intgroup = "group", ntop = 1000, returnData = TRUE)
PCs2
```

```
##          PC1          PC2 group group.1 name
## A1 -20.86001 -16.00295746     A       A   A1
## B1  10.80363 -15.48602457     B       B   B1
## A3 -16.34810  11.43224647     A       A   A3
## B2  12.84923   8.06337251     B       B   B2
## A4 -18.30490   8.11360291     A       A   A4
## B3  14.99738   0.07179807     B       B   B3
## B4  16.86277   3.80796208     B       B   B4
```

```
dds2 = DESeq(dds2)
```

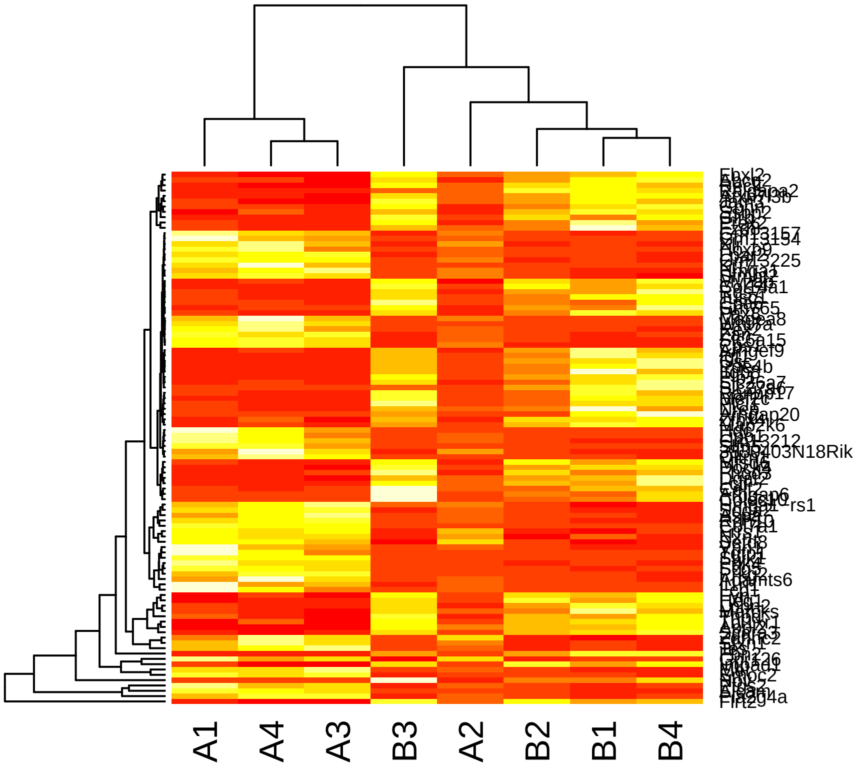
```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
DE_Results2 = results(dds2)
DE_Results2[DE_Results2$padj < 0.01 & !is.na(DE_Results2$padj),]
```

```
## log2 fold change (MAP): group B vs A
## Wald test p-value: group B vs A
## DataFrame with 1585 rows and 6 columns
##            baseMean log2FoldChange      lfcSE      stat       pvalue
##           <numeric>      <numeric>  <numeric> <numeric>    <numeric>
## Sgk3      421.10909      0.8941662  0.2497850  3.579743 3.439324e-04
## Cpa6       49.53204      1.8759462  0.4074570  4.604035 4.143825e-06
## Prex2     731.47316      2.1247903  0.4147767  5.122733 3.011392e-07
## Rdh10    1137.86780     -1.7955958  0.1941614 -9.247953 2.288324e-20
## Defb41     21.12648     -2.2140868  0.4164094 -5.317091 1.054394e-07
## ...             ...            ...        ...       ...          ...
## Nhlrc2    1426.8662      0.2917367 0.08085278  3.608246 3.082747e-04
## Afap1l2    363.5608      1.1401178 0.17697915  6.442102 1.178297e-10
## Ablim1    3087.9009      0.4146745 0.12212500  3.395492 6.850539e-04
## Atrnl1    1276.6648      0.8579542 0.22640695  3.789434 1.509910e-04
## Hspa12a    123.2419     -1.9962492 0.41659620 -4.791808 1.652847e-06
##                 padj
##            <numeric>
## Sgk3    3.697075e-03
## Cpa6    7.828347e-05
## Prex2   7.404733e-06
## Rdh10   3.987977e-18
## Defb41  2.821566e-06
## ...              ...
## Nhlrc2  3.375250e-03
## Afap1l2 5.909287e-09
## Ablim1  6.609703e-03
## Atrnl1  1.837344e-03
## Hspa12a 3.427156e-05
```

We now get almsot a four-fold increase in the number of "significant" genes. This makes sense considering we removed the case where a "B-like" sample was labelled A; the two groups are now linearly seperable in the projection onto the first two principle components, whereas before they were not.

```
ddsshort = dds[order(DE_Results$padj)[c(1:100)],]
heatmap(counts(ddsshort, normalized = TRUE))
```



This presents clustering similar to what was seen in the PCA. However, since we are choosing genes that we already know identify group A from group B, the result of the clustering algorithm is biased to our selection of data fed to it.

We know that genes with low expression levels tend not the distinguish between groups well. Since this is true regardless of the specific phenotype we are looking at, a less biased sampling would be to select for genes with high expression levels.