Differential Expression with DESeq2

Drosophila melanogaster

This document and the data used in this example can be found at:

```
https://software.rc.fas.harvard.edu/ngsdata/workshops/2015_March
```

or on the cluster at:

```
/n/ngsdata/workshops/2015_March
```

1. Setup

First, install DESeq2 (http://bioconductor.org/packages/release/bioc/html/DESeq2.html):

```
source('http://bioconductor.org/biocLite.R')
biocLite('DESeq2')
```

Then load the libraries we'll need into R:

```
library('DESeq2')
library('RColorBrewer')
```

---

2. Read gene counts into a *data frame*

Read sample gene counts a tab-delimited file. The rows of the data frame are genes while the columns are samples.

```
sampleNames <- c('dmel_unf1', 'dmel_unf2', 'dmel_unf3', 'dmel_inf1', 'dmel_inf2', 'dmel_inf3')
filePath = 'http://software.rc.fas.harvard.edu/ngsdata/workshops/2015_March/fruitfly.gene_counts.allsamples.tsv'
countData = read.table(file = filePath, header = TRUE, row.names = 1, sep = '\t')
dim(countData)  #view number of rows and columns
```

```
## [1] 17321     6
```

Now create a second data frame for sample information, such as the experimental condition that each sample belongs to:

```
condition <- c('Control', 'Control', 'Control', 'Infected', 'Infected', 'Infected')  #vector of column names for the data frame
colData <- data.frame(row.names=colnames(countData), condition=factor(condition, levels=c('Control','Infected')))
colData
```

```
##           condition
## dmel_unf1   Control
## dmel_unf2   Control
## dmel_unf3   Control
## dmel_inf1  Infected
## dmel_inf2  Infected
## dmel_inf3  Infected
```

---

3. Run DESeq2

First, create a DESeqDataSet by specifying the gene counts data frame, the sample information data frame and a design model:

```
dataset <- DESeqDataSetFromMatrix(countData = countData,
                                  colData = colData,
                                  design = ~condition)
dataset
```

```
## class: DESeqDataSet
## dim: 17321 6
## exptData(0):
## assays(1): counts
## rownames(17321): FBgn0000003 FBgn0000008 ... FBgn0267794
##   FBgn0267795
## rowData metadata column names(0):
## colnames(6): dmel_unf1 dmel_unf2 ... dmel_inf2 dmel_inf3
## colData names(1): condition
```

Then run the DESeq2 algorithm and extract results for our two-class comparison:

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

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

```
result <- results(dds, contrast=c('condition','Infected','Control'))
result <- result[complete.cases(result),]  #remove any rows with NA
head(result)
```

```
## log2 fold change (MAP): condition Infected vs Control
## Wald test p-value: condition Infected vs Control
## DataFrame with 6 rows and 6 columns
##              baseMean log2FoldChange     lfcSE        stat     pvalue
##             <numeric>      <numeric> <numeric>   <numeric>  <numeric>
## FBgn0000003  152.4314    0.003761435 0.2188509  0.01718721 0.98628727
## FBgn0000008  467.3243   -0.052537432 0.1221521 -0.43009843 0.66712404
## FBgn0000014  274.6920   -0.285326344 0.1571362 -1.81579014 0.06940257
## FBgn0000015  119.2080   -0.188844665 0.1729476 -1.09191834 0.27486900
## FBgn0000017 2258.3067    0.093765456 0.1462254  0.64123926 0.52136724
## FBgn0000018  282.8917   -0.041435638 0.1391776 -0.29771776 0.76591859
##                  padj
##             <numeric>
## FBgn0000003 0.9980025
## FBgn0000008 0.9290130
## FBgn0000014 0.4869521
## FBgn0000015 0.7585526
## FBgn0000017 0.8849077
## FBgn0000018 0.9517331
```

4. View results

View a summary of DESeq2 results:

```
summary(result)
```

```
##
## out of 10255 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)     : 199, 1.9%
## LFC < 0 (down)   : 202, 2%
## outliers [1]     : 0, 0%
## low counts [2]   : 0, 0%
## (mean count < 16.2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

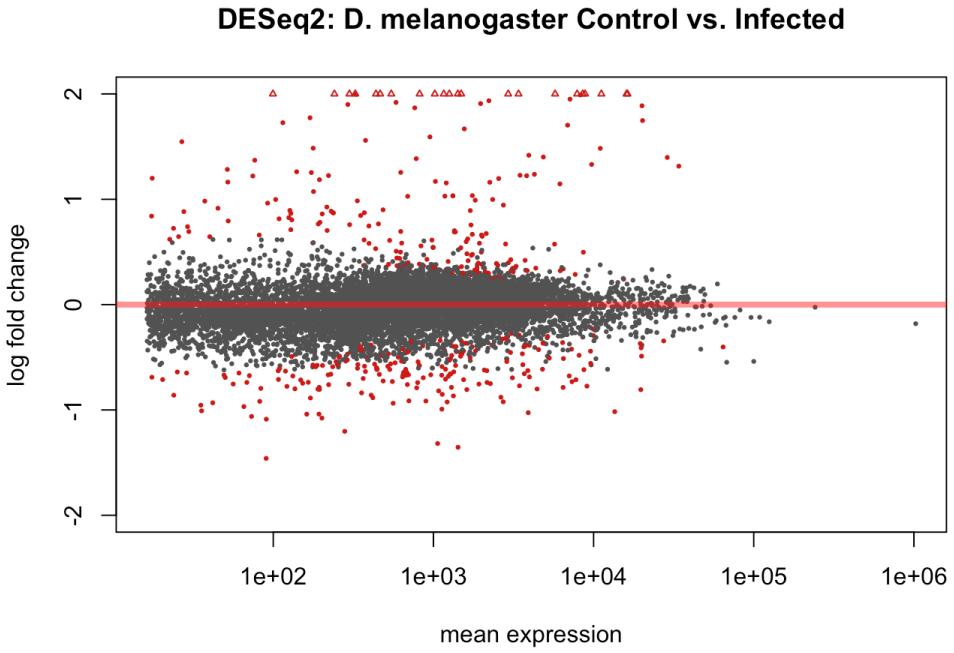
The top 50 up-regulated and down-regulated genes by p-value:

```
n = 50
resOrdered <- result[order(result$padj),]
topResults <- rbind( resOrdered[ resOrdered[,'log2FoldChange'] > 0, ][1:n,],
                     resOrdered[ resOrdered[,'log2FoldChange'] < 0, ][n:1,] )
topResults[c(1:5,(2*n-4):(2*n)), c('baseMean','log2FoldChange','padj')]
```

```
## DataFrame with 10 rows and 3 columns
##                 baseMean log2FoldChange          padj
##                <numeric>      <numeric>     <numeric>
## FBgn0041579  3398.68174      5.4624070 8.588424e-168
## FBgn0041581 11178.54200      4.7923385 4.177076e-116
## FBgn0000279  1490.59149      4.6903996 4.062114e-107
## FBgn0012042  8844.02836      4.5298778  4.415571e-98
## FBgn0262881   819.81287      4.5347213  2.092458e-97
## FBgn0027571 13552.37909     -1.0167664  5.643665e-08
## FBgn0266406    90.24966     -1.4599927  1.691359e-08
## FBgn0032297   560.37067     -0.9365595  1.085821e-08
## FBgn0052647   702.87904     -0.9139778  3.371036e-11
## FBgn0033446  3906.65774     -1.0260470  5.970731e-16
```

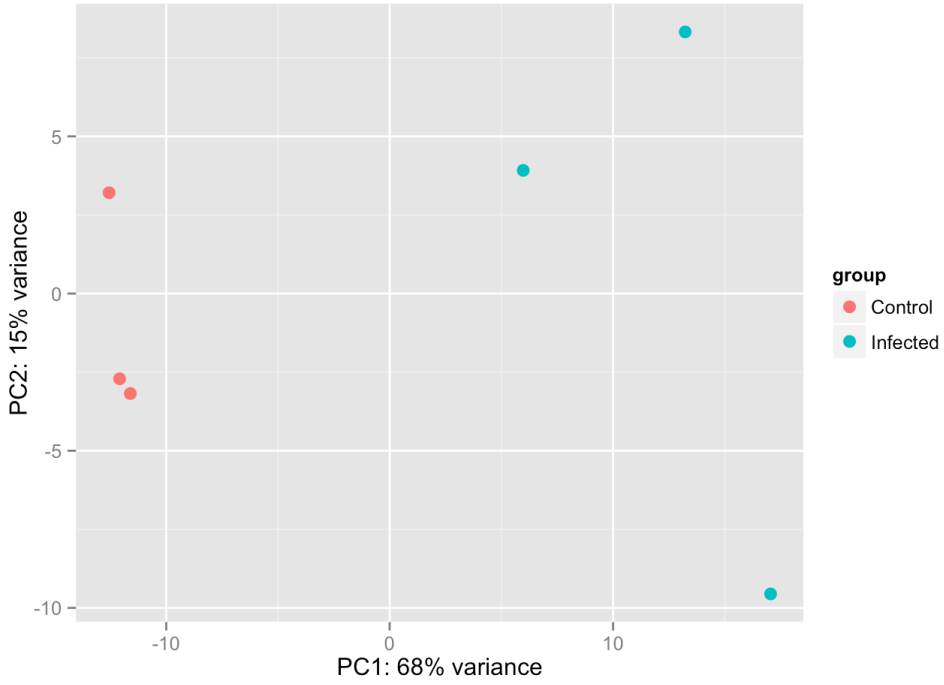
Plot log fold change vs. mean expression for all genes, with genes where p < 0.1 colored red:

```
plotMA(result, main='DESeq2: D. melanogaster Control vs. Infected', ylim=c(-2,2))
```



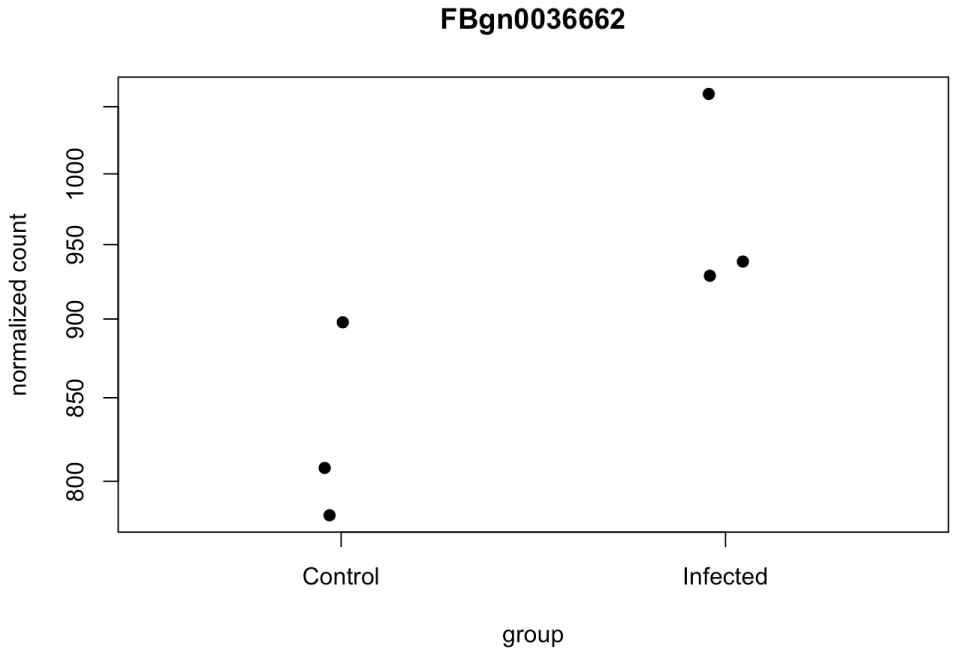
PCA plot for all genes:

```
rld <- rlogTransformation(dds, blind=TRUE)
plotPCA(rld)
```



Plot counts for a single gene. Below is the plot for the gene with the lowest p-value:

```
plotCounts(dds, gene=which.min(result$padj), intgroup='condition', pch = 19)
```



Display top genes' normalized counts in a heatmap:

```
hmcol <- brewer.pal(11,'RdBu')
nCounts <- counts(dds, normalized=TRUE)
heatmap(as.matrix(nCounts[ row.names(topResults), ]), Rowv = NA, col = hmcol, mar = c(8,2))
```