Differential Expression with DESeq2

Mouse immune cells

Control vs. treatment samples

This document and the data 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 from a tab-delimited file into a data frame. The rows of the data frame are genes while the columns are samples.

```
countFilePath = 'http://software.rc.fas.harvard.edu/ngsdata/workshops/2015_March/NC11.gene.txt'
countData = read.table(file = countFilePath, header = TRUE, sep = '\t', row.names = 1)
countData = countData[3:ncol(countData)]  #discard chr and strand columns
dim(countData)
```

```
## [1] 16241    24
```

Read in a second data frame that contains the experimental condition that each sample belongs to:

```
colFilePath = 'http://software.rc.fas.harvard.edu/ngsdata/workshops/2015_March/NC11.colData_2condi
tions.txt'
colData = read.table(file = colFilePath, header = TRUE, sep = '\t', row.names = 1)
colData[['condition']] =  factor(colData[['condition']], levels = c('Control', 'Treatment'))

colData
```

```
##                      condition
## HMW_4h_NC11          Treatment
## LPS.PAM.HMW_2h_NC11 Treatment
## LPS.PAM_6h_NC11      Treatment
## LPS.PAM_4h_NC11      Treatment
## Ctrl2_NC11             Control
## LPS.HMW_4h_NC11      Treatment
## PAM.HMW_4h_NC11      Treatment
## LPS.PAM.HMW_4h_NC11 Treatment
## LPS.PAM_2h_NC11      Treatment
## Ctrl3_NC11             Control
## HMW_6h_NC11          Treatment
## PAM_4h_NC11          Treatment
## LPS_2h_NC11          Treatment
## LPS.HMW_6h_NC11      Treatment
## LPS.PAM.HMW_6h_NC11 Treatment
## HMW_2h_NC11          Treatment
## PAM_2h_NC11          Treatment
## Ctrl1_NC11             Control
## LPS_4h_NC11          Treatment
## PAM.HMW_2h_NC11      Treatment
## PAM_6h_NC11          Treatment
## LPS.HMW_2h_NC11      Treatment
## PAM.HMW_6h_NC11      Treatment
## LPS_6h_NC11          Treatment
```

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)
```

```
## converting counts to integer mode
```

```
dataset
```

```
## class: DESeqDataSet
## dim: 16241 24
## exptData(0):
## assays(1): counts
## rownames(16241): Ppp1r14c Plekhg1 ... Samd11 Vamp7
## rowData metadata column names(0):
## colnames(24): HMW_4h_NC11 LPS.PAM.HMW_2h_NC11 ... PAM.HMW_6h_NC11
##   LPS_6h_NC11
## 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
## -- replacing outliers and refitting for 568 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
## estimating dispersions
## fitting model and testing
```

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

```
## log2 fold change (MAP): condition Treatment vs Control
## Wald test p-value: condition Treatment vs Control
## DataFrame with 6 rows and 6 columns
##                baseMean log2FoldChange     lfcSE       stat       pvalue
##               <numeric>      <numeric> <numeric>  <numeric>    <numeric>
## Plekhg1       45.418246      0.9394347 0.4048746  2.3203103 2.032410e-02
## Mthfd1l       32.991165     -0.4364079 0.3681590 -1.1853787 2.358677e-01
## 1700052N19Rik 14.459580     -0.6506307 0.4223172 -1.5406210 1.234091e-01
## Esr1           9.722313      0.5018282 0.5451788  0.9204838 3.573200e-01
## Oprm1         38.339219     -0.1952093 0.2726131 -0.7160670 4.739500e-01
## Lrp11         46.353306      2.0091578 0.5075967  3.9581775 7.552381e-05
##                      padj
##                 <numeric>
## Plekhg1       0.09733541
## Mthfd1l       0.44060763
## 1700052N19Rik 0.29524410
## Esr1          0.56360262
## Oprm1         0.66636793
## Lrp11         0.00224374
```

4. View results

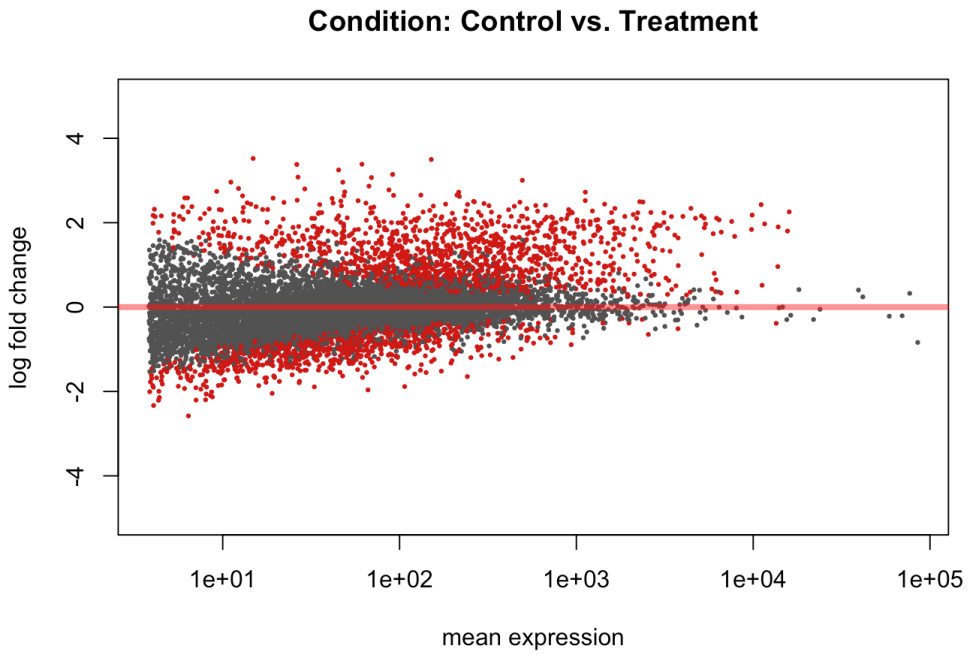
A summary of DESeq2 results:

```
summary(result)
```

```
##
## out of 10517 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)     : 1358, 13%
## LFC < 0 (down)   : 862, 8.2%
## outliers [1]     : 0, 0%
## low counts [2]   : 0, 0%
## (mean count < 3.8)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

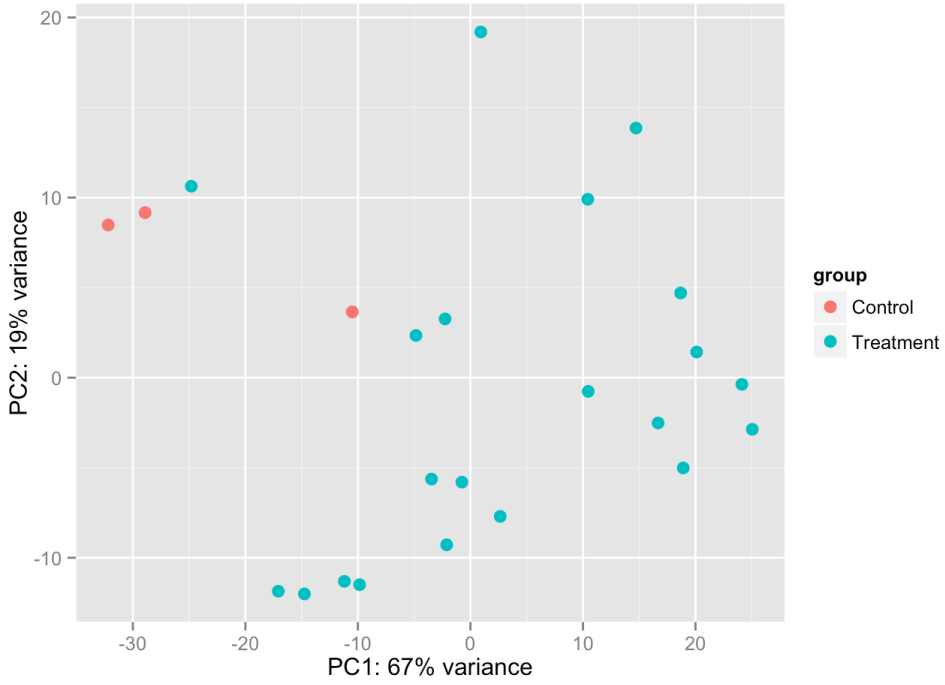
Plot log fold change vs. mean expression for all genes, with genes where p < 0.1 colored red:

```
plotMA(result, main=paste0('Condition: Control vs. Treatment'), ylim=c(-5,5))
```



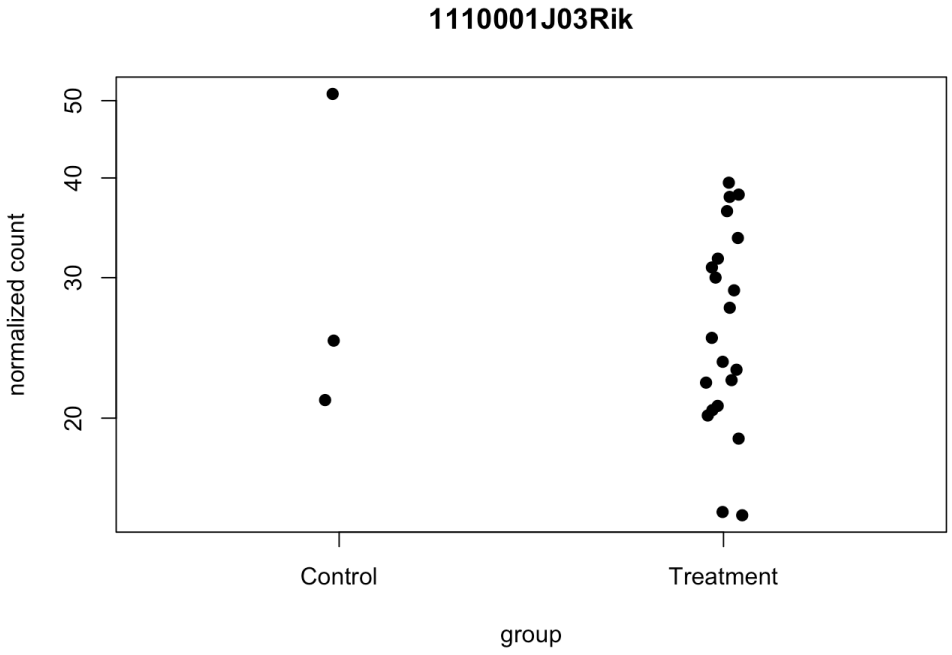
PCA plot for all genes:

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



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)
```



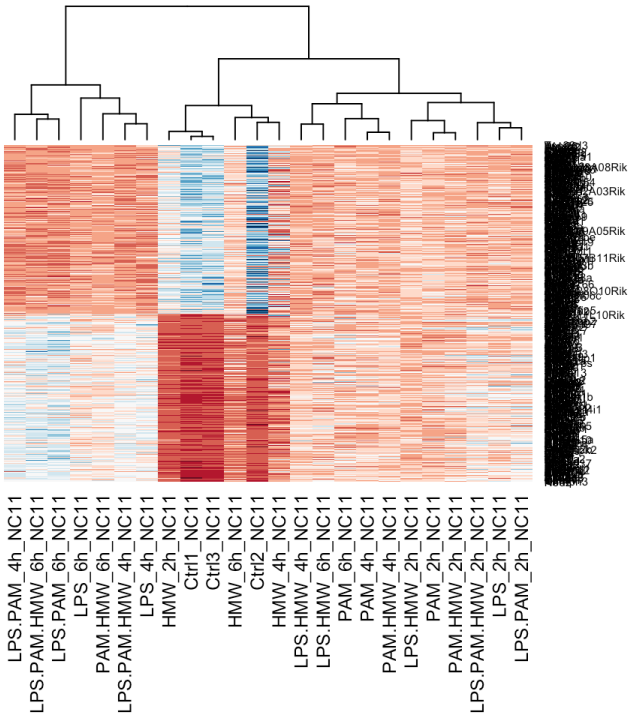
Extract results for the top 250 up-regulated and top 250 down-regulated genes, sorted by p-value:

```
n = 250
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')]  #print results for top an
d bottom 5 genes
```

```
## DataFrame with 10 rows and 3 columns
##           baseMean log2FoldChange         padj
##          <numeric>      <numeric>    <numeric>
## Nod2      91.36978      3.1440017 3.468863e-13
## Hnrnph3 257.79080      1.6069051 3.468863e-13
## Olr1    536.67525      1.8926661 4.263801e-09
## Daam1   158.62200      2.1623200 6.784224e-09
## Epha4   151.09879      3.4976248 3.186886e-08
## Ptcd2    40.74538     -1.2707074 5.940612e-04
## Lonrf3  111.19039     -1.4381939 5.722514e-04
## Ptp4a2  950.81454     -0.5278149 2.082919e-04
## Wdr26   483.09454     -0.4878060 9.595422e-05
## Tsc22d3  19.02615     -2.0443879 5.886623e-05
```

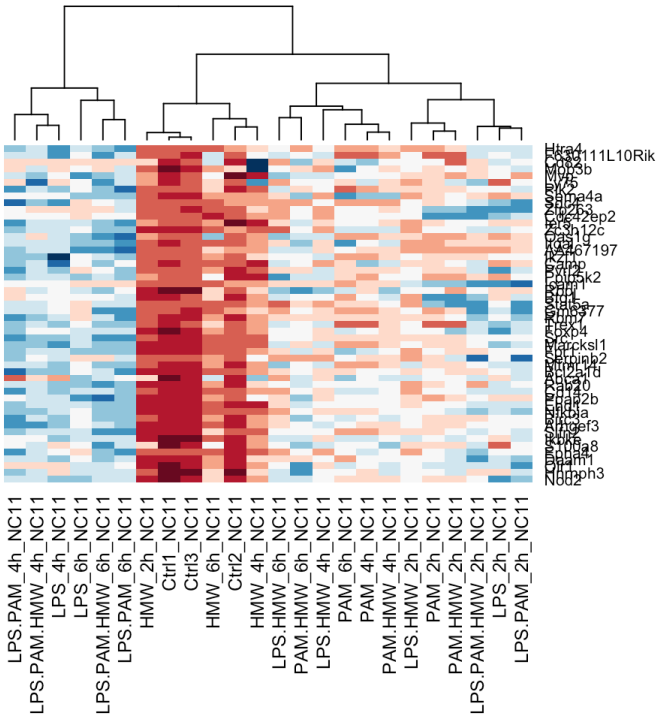
Display these top genes' normalized counts in a heatmap, and cluster samples by similarity:

```
hmcol <- brewer.pal(11,'RdBu')
nCounts <- counts(dds, normalized=TRUE)
heatmap(as.matrix(nCounts[ row.names(topResults), ]), Rowv = NA, col = hmcol, mar = c(10,2))
```



Examine sample clusters that arise from the top 25 and bottom 25 genes are used:

```
m = 25
heatmap(as.matrix(nCounts[ row.names(topResults)[c(1:m,(n-m+1):n)], ]), Rowv = NA, col = hmcol, ma
r = c(10,2))
```



Note the similarities and differences in the sample clusters that occur when using only the very top up and down genes, verses using a broader representation of each sample.

5. Write DESeq2 data to file

```
project.dir <- '~/My_R_Example'
dir.create(project.dir, showWarnings=FALSE)
write.table(result, file = file.path(project.dir,paste0('NC11_Control_vs_Treatment.tsv')), quote =
 FALSE, sep = '\t')
```