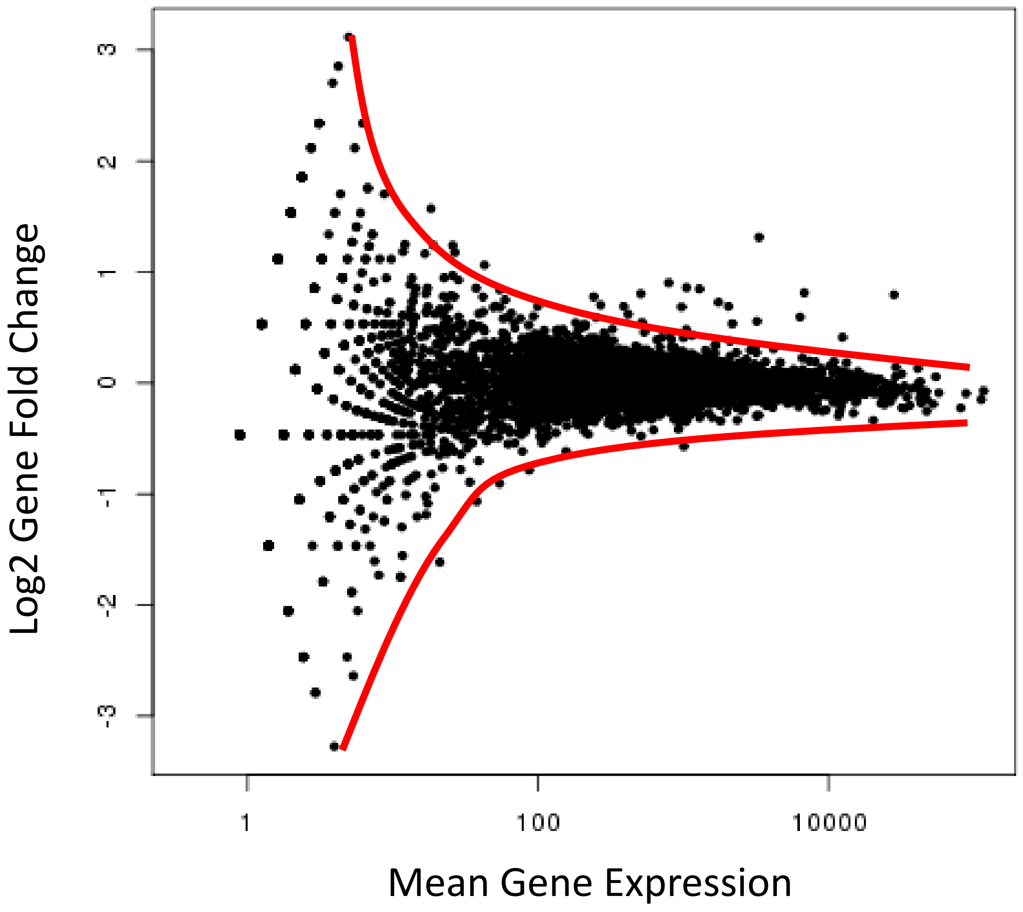
Day 7

RNA-seq and Differential Expression Analysis

Working with DESeq

What is DESeq?

- DESeq is an R package
- provides methods to test for differential expression by use of the negative binomial distribution
- For any given gene DESeq estimates the mean (μ) expression of that gene and the variance in expression (σ^2)
  - assumes that expression of a given gene across multiple samples can be modeled by a negative binomial (NB) distribution



Log2 Gene Fold Change
3
2
1
0
-1
-2
-3
1
100
10000
Mean Gene Expression

HTSeq Counts

Output from HTSeq counts for each sample

HTSeq Counts

Gene	Counts
HRA1	0
LSR1	421
NME1	636
RDN18-1	0
RDN18-2	0
RDN25-1	0
RDN25-2	0

DESeq Counts Table

Gene	Counts 1	Counts 2	Counts 3
HRA1	0	5	0
LSR1	421	500	400
NME1	636	600	674
RDN18-1	0	0	0
RDN18-2	0	0	0
RDN25-1	0	0	0
RDN25-2	0	0	0

```
$ paste <all HTSeq count files> > allCounts.txt
$ cut -f1,2,4,6,8 allCounts.txt > DESeq.counts.txt
```

Working with R

- DESeq is an R package
- R is an open source software environment for statistical computing and graphics
  - Operates with command lines and jobscripts
  - Has its own language and commands

```
$ qsub –I
–W x=FLAGS:ADVRES:sreadreslrg.0.0
–l walltime=3:00:00 –l nodes=1:ppn=1
–q short

$ /opt/R/2.15.1/bin/R
```

DESeq in R

- Terminal now is R, not unix
- To clear R console > ctrl+l (= unix clear command)

- Load DESeq package

```
> library(DESeq)
```

- List all DESeq functions

```
> ls(pos="package:DESeq")
```

- Load counts data into R variable called data

```
> data = read.delim(
“/projects/sreadgrp/Day7/DESeq.counts.txt”, sep=“\t”,
header=TRUE, row.names=1)
```

Loading Gene Counts in R

```
> data =
read.delim("/projects/sreadgrp/Day7/DESeq.counts.txt"
, sep="\t", header=TRUE, row.names=1)

> head(data)
          WT.1 WT.2 RRP6.1 RRP6.2
HRA1         0    0      3      0
LSR1       164  313   1610   2207
NME1       422  652   5060   7234
RDN18-1      0    0      0      0
RDN18-2      0    0      0      0
RDN25-1      0    0      0      0
```

Setting up CDS (<u>c</u>ount <u>d</u>ata <u>s</u>et)

```
> conditions = c("WT", "WT", "RRP6", "RRP6")
> cds = newCountDataSet(data, conditions)
> cds = estimateSizeFactors(cds)
> sizeFactors(cds)
WT.1      WT.2      RRP6.1    RRP6.2
0.813092  0.932020  0.974623  1.3471957
> ?estimateDispersions
```

Select 1

Brings up usage information for function estimateDispersions

estimateDispersions

estimateDispersions(

method – how to calculate variance estimates to counts (default = ‘pooled’)

sharingMode – how to apply variance to gene expression value (default = “maximum”)

)

estimateDispersions

estimateDispersions(method=

'**pooled**' – estimate one pooled dispersion estimate across all replicated samples (default), robost against outliers

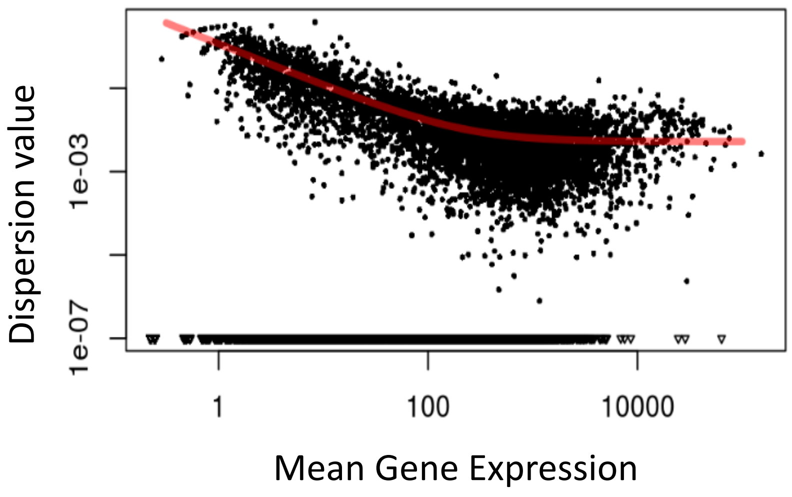
'**per-condition**' - For each condition with replicates, compute a gene's empirical dispersion value by considering the data from samples for this condition. For samples of unreplicated conditions, the maximum of empirical dispersion values from the other conditions is used.

'**blind**' - no replicates, estimate dispersion across all samples

estimateDispersions

estimateDispersions(sharingMode=

'**fit-only**' - use only the fitted value. Use this only with very few replicates, and when you are not too concerned about false positives from dispersion outliers



'**maximum**' - take the maximum of the two values. Recommended if you have at least 3 or 4 replicates

'**gene-est-only**' - No fitting or sharing, use only the empirical value. This method is preferable when the number of replicates is large and the empirical dispersion values are sufficiently reliable (or else you have a high false positive rate

```
> cds = estimateDispersions(cds, method='pooled',
sharingMode='fit-only')

> results = nbinomTest(cds, "WT", "RRP6")

> write.table(results,
file="/Users/verajm/DESeq.results.txt", sep="\t",
row.names=FALSE)

> significant =  results[results$padj < 0.1,]
```

Notes on R

To exit R: > quit() then select “n”

You can run R jobscripts (called Rscripts, file.R) from the Unix command line

```
$ Rscript file.R
```