

Class 12: Transcriptomics and the analysis of RNA-Seq data

Angela Bartolo PID: A15932451

Here we will use the DESeq2 package for RNASeq analysis. The data for today's class comes from a study of airway smooth muscle cells were treated with dexamethasone, a synthetic glucocorticoid steroid with anti-inflammatory effects (Himes et al. 2014).

Import thir data

We need two things for this analysis:

- countData (counts for every transcript/gene in each experiment)
- colData (metadata that describes the experimental setup)

```
countData <- read.csv("airway_scaledcounts.csv", row.names=1)
head(countData)
```

	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516
ENSG000000000003	723	486	904	445	1170
ENSG000000000005	0	0	0	0	0
ENSG000000000419	467	523	616	371	582
ENSG000000000457	347	258	364	237	318
ENSG000000000460	96	81	73	66	118
ENSG000000000938	0	0	1	0	2
	SRR1039517	SRR1039520	SRR1039521		
ENSG000000000003	1097	806	604		
ENSG000000000005	0	0	0		
ENSG000000000419	781	417	509		
ENSG000000000457	447	330	324		
ENSG000000000460	94	102	74		
ENSG000000000938	0	0	0		

```
metadata <- read.csv("airway_metadata.csv")
head(metadata)
```

```
    id      dex celltype     geo_id
1 SRR1039508 control   N61311 GSM1275862
2 SRR1039509 treated   N61311 GSM1275863
3 SRR1039512 control   N052611 GSM1275866
4 SRR1039513 treated   N052611 GSM1275867
5 SRR1039516 control   N080611 GSM1275870
6 SRR1039517 treated   N080611 GSM1275871
```

Q1. How many genes are in this dataset?

```
nrow(countData)
```

```
[1] 38694
```

Q2. How many ‘control’ cell lines do we have?

```
sum( metadata$dex == "control" )
```

```
[1] 4
```

```
# Another way to solve
# table(metadata$dex)
```

Finding means of control and treated

- Step 1

Calculate the mean of the control samples (i.e. columns in the countData)

- We need to find which columns in the countData are “control” samples
- Extract all the control columns from countData and call it `control.counts`
- Calculate the mean value across the rows of `control.counts` i.e calculate the mean count values for each gene in the control samples.

```
#a
control inds <- metadata$dex == "control"

#b
```

```

control.counts <- countData[ , control inds ]
#c
# Another way to solve
# apply( control.counts , 1 , mean )

control.means <- rowMeans(control.counts)
head(control.means)

```

```

ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
      900.75          0.00       520.50       339.75       97.25
ENSG000000000938
      0.75

```

- Step 2

Calculate the mean of the treated samples

```

treated.means <- rowMeans( countData[ , metadata$dex == "treated" ] )
head(treated.means)

```

```

ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
      658.00          0.00       546.00       316.50       78.75
ENSG000000000938
      0.00

```

We now have control and treated mean count values. For ease of book-keeping I will combine these vectors into a new data.frame called **meancounts**

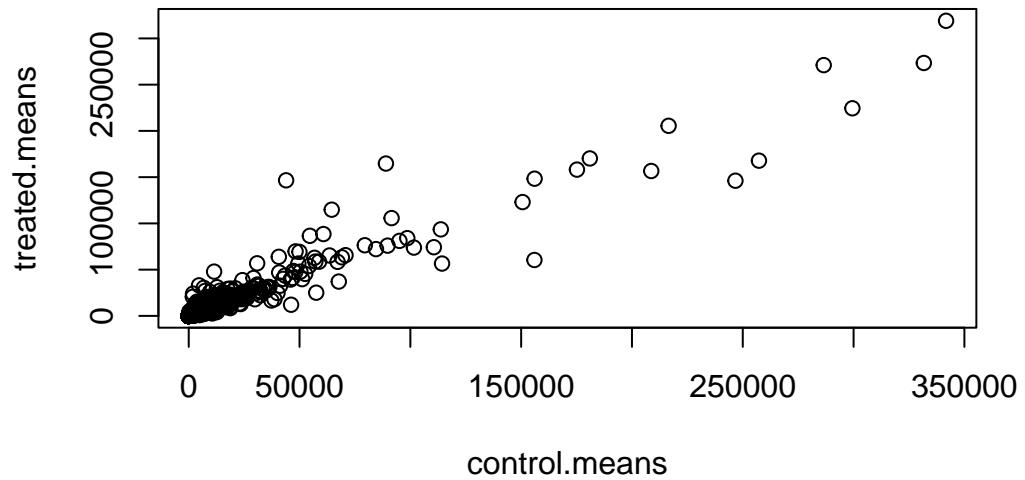
```

meancounts <- data.frame(control.means, treated.means)
head(meancounts)

```

	control.means	treated.means
ENSG00000000003	900.75	658.00
ENSG00000000005	0.00	0.00
ENSG00000000419	520.50	546.00
ENSG00000000457	339.75	316.50
ENSG00000000460	97.25	78.75
ENSG000000000938	0.75	0.00

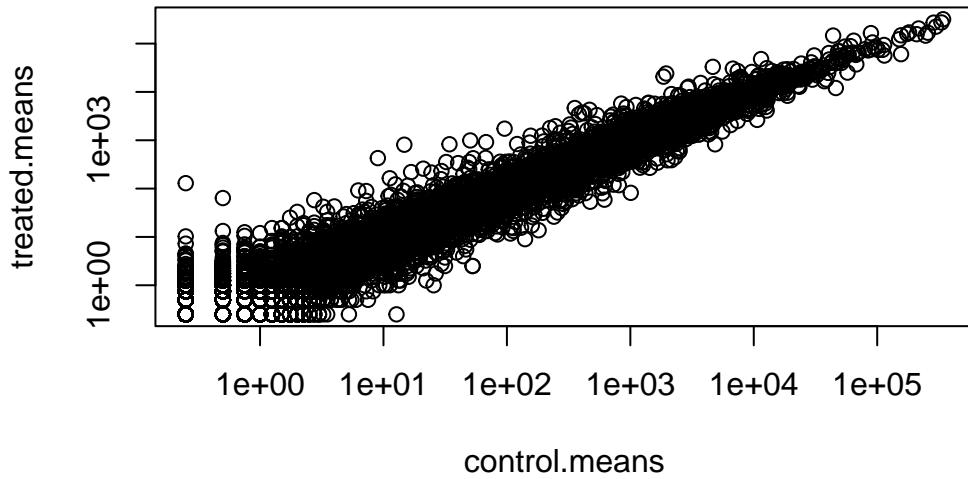
```
plot(meancounts)
```



```
plot(meancounts, log = "xy")
```

Warning in xy.coords(x, y, xlabel, ylabel, log): 15032 x values <= 0 omitted
from logarithmic plot

Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted
from logarithmic plot



We use logtransforms for skewed data such as this and because we really care most about relative changes in magnitude.

We most often use log2 as our transform as the math is easier to interpret than log10 or other.

If we have no change - i.e. same values in the control and treated we will have a log2 values of zero

```
log2(20/20)
```

```
[1] 0
```

If I have double the amount i.e. 20 compared to 10, I will have a log2 fold-change of +1 (increase)

```
log2(20/10)
```

```
[1] 1
```

If I have half the amount i.e. 10 compared to 20, I will have a log2 fold-change of -1 (decrease)

```
log2(10/20)
```

```
[1] -1
```

```
meancounts$log2fc <- log2(meancounts$treated.means / meancounts$control.means)
head(meancounts)
```

	control.means	treated.means	log2fc
ENSG000000000003	900.75	658.00	-0.45303916
ENSG000000000005	0.00	0.00	NaN
ENSG000000000419	520.50	546.00	0.06900279
ENSG000000000457	339.75	316.50	-0.10226805
ENSG000000000460	97.25	78.75	-0.30441833
ENSG000000000938	0.75	0.00	-Inf

Q. How many genes are upregulated at the common threshold of +2 log2fc values

```
sum(meancounts$log2fc >=2, na.rm=T)
```

```
[1] 1910
```

What about the stats (p-value). They are big changes but are they statistically significant

To do this properly we will turn to DESeq2 package

DESeq2 analysis

```
library(DESeq2)
```

To use DESeq we need out input countData and colData in a specific formate that DESeq wants:

```
dds <- DESeqDataSetFromMatrix(countData = countData,
                                colData = metadata,
                                design = ~dex)
```

```
converting counts to integer mode
```

```
Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in  
design formula are characters, converting to factors
```

To run the analysis I can now use the main DESeq2 function called `DESeq` with `dds` as input.

```
  dds <- DESeq(dds)  
  
  estimating size factors  
  
  estimating dispersions  
  
  gene-wise dispersion estimates  
  
  mean-dispersion relationship  
  
  final dispersion estimates  
  
  fitting model and testing
```

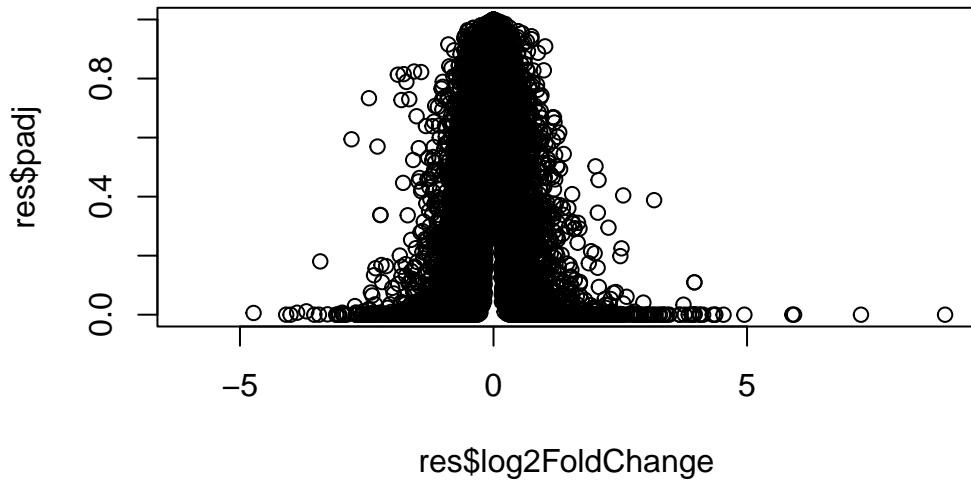
To get the results out of this `dds` object we can use the `results()` function from the package.

```
  res <- results(dds)  
  head(res)  
  
  log2 fold change (MLE): dex treated vs control  
  Wald test p-value: dex treated vs control  
  DataFrame with 6 rows and 6 columns  
    baseMean log2FoldChange      lfcSE      stat     pvalue  
    <numeric>      <numeric> <numeric> <numeric> <numeric>  
  ENSG000000000003 747.194195 -0.3507030  0.168246 -2.084470 0.0371175  
  ENSG000000000005  0.000000   NA        NA        NA        NA  
  ENSG00000000419  520.134160  0.2061078  0.101059  2.039475 0.0414026  
  ENSG00000000457  322.664844  0.0245269  0.145145  0.168982 0.8658106  
  ENSG00000000460  87.682625 -0.1471420  0.257007 -0.572521 0.5669691  
  ENSG00000000938  0.319167 -1.7322890  3.493601 -0.495846 0.6200029  
    padj  
    <numeric>  
  ENSG000000000003  0.163035  
  ENSG000000000005   NA
```

```
ENSG00000000419  0.176032  
ENSG00000000457  0.961694  
ENSG00000000460  0.815849  
ENSG00000000938      NA
```

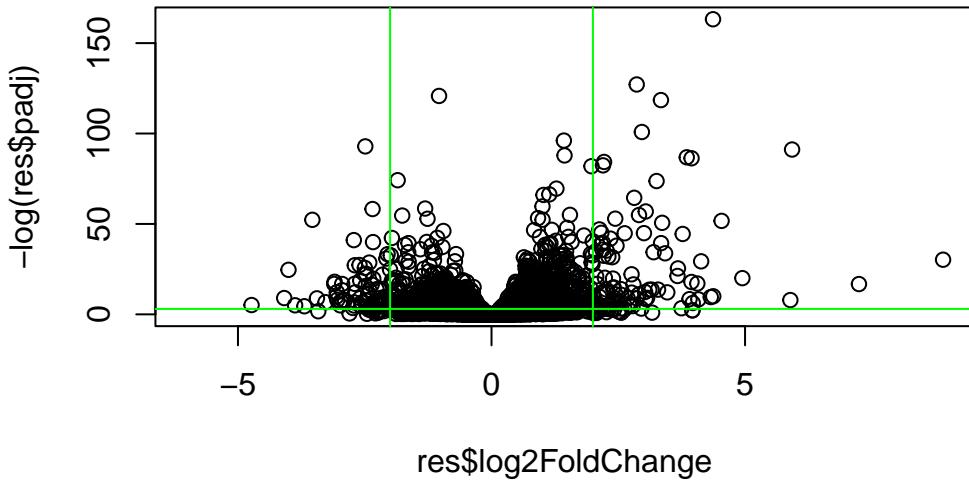
Let's makes a final (for today) plot of log2 fold-chain vs the adjusted P-values.

```
plot(res$log2FoldChange, res$padj)
```



It is the low p-values that we care about and these are lost in the skewed plot above. Let's take the log of the \$padj values for our plot.

```
plot(res$log2FoldChange, -log(res$padj))  
abline(v=c(+2,-2), col = "green")  
abline(h= -log(0.05), col = "green")
```



Finally we can make a color vector to use in the plot to better highlight the genes we care about.

```

mycols <- rep("grey", nrow(res))
mycols[abs(res$log2FoldChange) >= 2] <- "red"
mycols[res$padj > 0.05] <- "grey"

plot(res$log2FoldChange, -log(res$padj), col=mycols)
abline(v=c(+2,-2), col = "green")
abline(h= -log(0.05), col = "green")

```

