

Class 12: RNASeq Analysis

Rachel Galleta (A16859649)

Table of contents

Background	1
Data Import	1
Toy differential gene expression	3
Volcano Plot	8
save our results	9

Background

Today we will analyze some RNASeq data from Himes et al. on the effects of a common steroid (dexamethasone,) on airway smooth muscle cells (ASm cells)

Are staring point is the “counts” data and “metadata” that contain the count values for each gene in their different experiments (i.e cell lines with or without drugs)

Data Import

```
counts <- read.csv("airway_scaledcounts.csv", row.names=1)
metadata <-read.csv("airway_metadata.csv")
```

let's have a wee peak at these objects:

```
head(counts)
```

	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		

Q. How many different experiments (columns in counts metadata) are there?

```
ncol(counts)
```

[1] 8

```
nrow(metadata)
```

[1] 8

Q1. How many genes are in this dataset?

```
nrow(counts)
```

[1] 38694

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

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

[1] 4

Toy differential gene expression

To start our analysis lets calculate the mean for all genes in the “control” experiments.

1. extract all “control” columns form the counts objets
2. Calculate the mean for all rows (ie, genes) of these “control” columns 3-4. Do the same for “treated”
3. Compare these “control.mean” and “treated.mean” values.

```
#1.  
control.ind <- metadata$dex=="control"  
control.counts<- counts[,control.ind]  
  
#2.  
control.means <- rowMeans( control.counts)  
  
dim(control.counts)
```

[1] 38694 4

```
#3-4.  
treated.ind <- metadata$dex=="treated"  
treated.counts<- counts[,treated.ind]  
treated.means <- rowMeans(treated.counts)
```

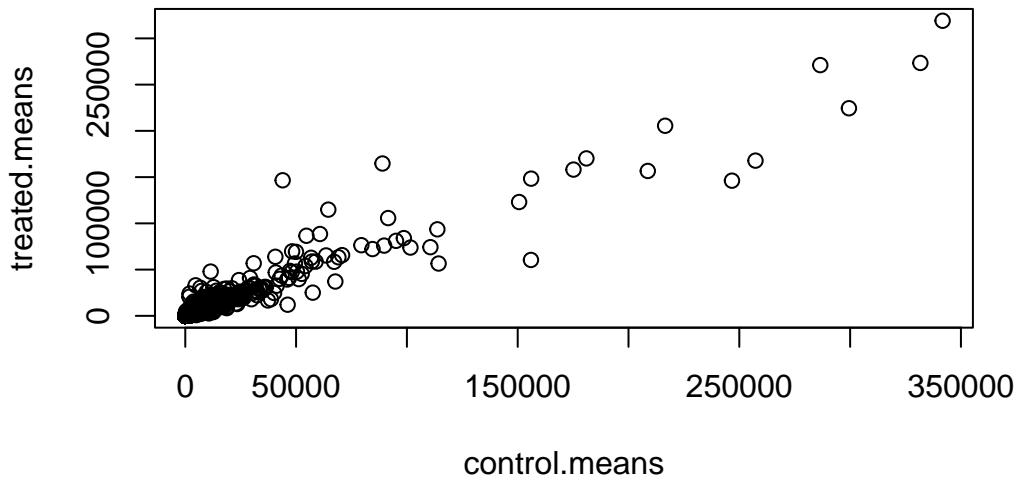
store these together for ease of book means counts

```
#5.  
meancounts <- data.frame(control.means, treated.means)  
head(meancounts)
```

	control.means	treated.means
ENSG000000000003	900.75	658.00
ENSG000000000005	0.00	0.00
ENSG000000000419	520.50	546.00
ENSG000000000457	339.75	316.50
ENSG000000000460	97.25	78.75
ENSG000000000938	0.75	0.00

mamke a plot onf ocntorl vs treatments menad valeues for all genes

```
plot(meancounts)
```

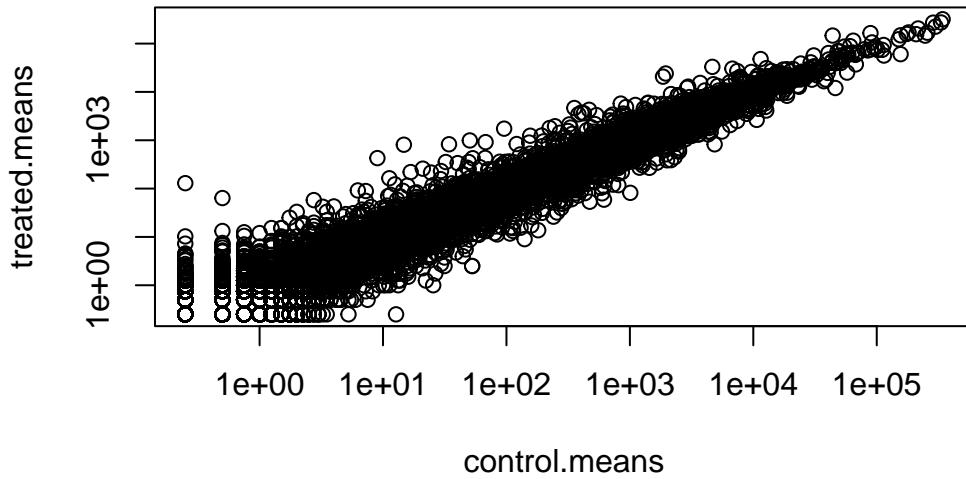


Make this a log plot

```
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 often talk about metrics like “log2 fold-change”

```
#treated/control
log2(10/10)
```

```
[1] 0
```

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

```
[1] -1
```

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

```
[1] 1
```

```
log2(40/10)
```

```
[1] 2
```

```
log2(10/40)
```

```
[1] -2
```

lets calculate the log2 fold change four our treated over control mean counts

```
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

A common “rule of thumb” is a log2 fold change cutoff of +2 and -2 to call genes “up regulated” or “down regulated”

Number of “up” genes

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

```
[1] 1846
```

number of “down” genes at -2 threshold

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

```
[1] 2330
```

##DESeq2 analysis

lest do this analysis properly and keep our inner starts nerd happy are the teh differenes we seen and no drug signifcnat givne the repliclate experimnet

```
library(DESeq2)
```

Warning: package 'IRanges' was built under R version 4.4.2

Warning: package 'GenomeInfoDb' was built under R version 4.4.2

Warning: package 'MatrixGenerics' was built under R version 4.4.2

Warning: package 'matrixStats' was built under R version 4.4.3

for DESeq analysis we need three things -count values ('countData') -metadata telling us about the columns in 'countData' ('colData') -design of the experiment (what do you want to compare)

our first function fromDESeq 2 will set up the input required for analysis by storing all these 3 things together.

```
dds<- DESeqDataSetFromMatrix(countData=counts,
                               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

the main function in DESeq2 that runs the analysis is called DESeq()

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

estimating size factors

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

```
final dispersion estimates
```

```
fitting model and testing
```

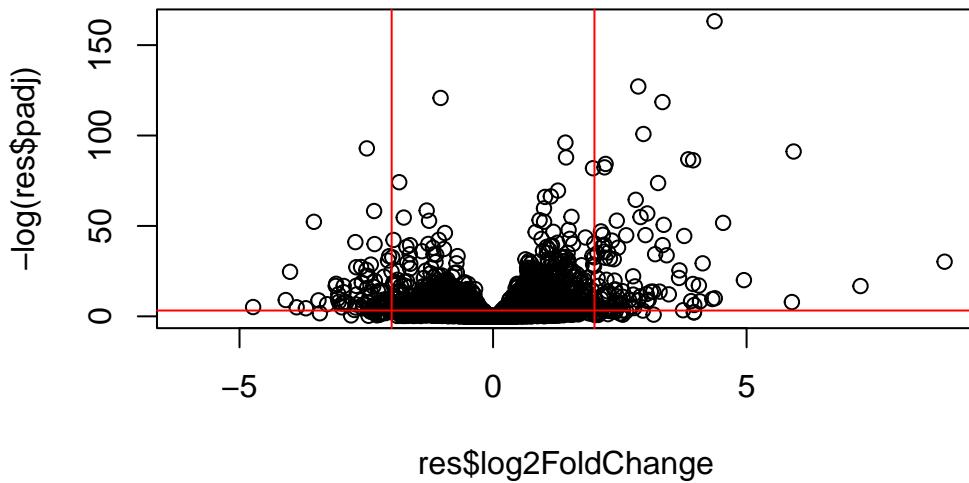
```
res<-results(dds)
head(res)
```

```
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>
ENSG00000000003 747.194195 -0.3507030 0.168246 -2.084470 0.0371175
ENSG00000000005  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>
ENSG00000000003  0.163035
ENSG00000000005   NA
ENSG00000000419  0.176032
ENSG00000000457  0.961694
ENSG00000000460  0.815849
ENSG00000000938   NA
```

Volcano Plot

this us common summary result fomr figrue thse tyoes of expreimnts and plot the log2 fold change vs the adjusted p-value

```
plot(res$log2FoldChange,-log(res$padj))
abline(v=c(-2,2),col="red")
abline(h=-log(0.04),col="red")
```



save our results

```
write.csv(res,file="my_results.csv")
```