Lab 12 RNAseq with DESeq2

Qingyun Zheng (A16338254)

Table of contents

Background

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

For this analysis we need two main inputs:

- `countData`: a table of counts per gene (in rows) accross experiments (in columns)
- `colData`: metadata about the design of the experiments. The rows match the columns in `countData`

Data Import

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

Let's take a look at the `counts` data

```
head(counts)
```

```
                SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
ENSG00000000003        723        486        904        445       1170
ENSG00000000005          0          0          0          0          0
ENSG00000000419        467        523        616        371        582
ENSG00000000457        347        258        364        237        318
ENSG00000000460         96         81         73         66        118
ENSG00000000938          0          0          1          0          2
                SRR1039517 SRR1039520 SRR1039521
ENSG00000000003       1097        806        604
ENSG00000000005          0          0          0
ENSG00000000419        781        417        509
ENSG00000000457        447        330        324
ENSG00000000460         94        102         74
ENSG00000000938          0          0          0
```

and the `metadata`:

```
metadata
```

```
               dex celltype     geo_id
SRR1039508 control   N61311 GSM1275862
SRR1039509 treated   N61311 GSM1275863
SRR1039512 control  N052611 GSM1275866
SRR1039513 treated  N052611 GSM1275867
SRR1039516 control  N080611 GSM1275870
SRR1039517 treated  N080611 GSM1275871
SRR1039520 control  N061011 GSM1275874
SRR1039521 treated  N061011 GSM1275875
```

> Q1. How many "genes" are in the dataset?

```
nrow(counts)
```

```
[1] 38694
```

> Q2. How many experiments (i.e. columns in `counts` or rows in `metadata`) are in the dataset?

```
ncol(counts)
```

```
[1] 8
```

```r
nrow(metadata)
```

```
[1] 8
```

> Q3. How many control experiments are in the dataset?

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

```
[1] 4
```

1. Extract the "control" columns from `counts`
2. Calculate the mean value for each gene in these "control columns" 3-4. Do the same for the "treated" columns
3. Compare the means

```r
#Step 1
ctrl.inds <- metadata$dex == "control"
ctrl.counts <- counts[, ctrl.inds]

#Step 2
ctrl.means <- rowMeans(ctrl.counts)

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

#Step 5
head(ctrl.means > treated.means)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460 
           TRUE           FALSE           FALSE            TRUE            TRUE 
ENSG00000000938 
           TRUE 
```

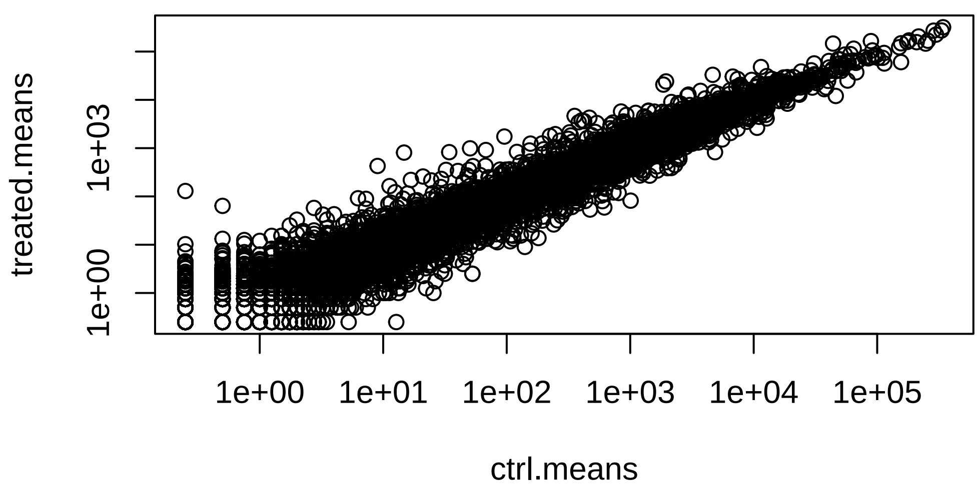
```r
meancounts <- data.frame(ctrl.means, treated.means)
head(meancounts)
```

```
                ctrl.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
ENSG00000000938       0.75          0.00
```

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

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

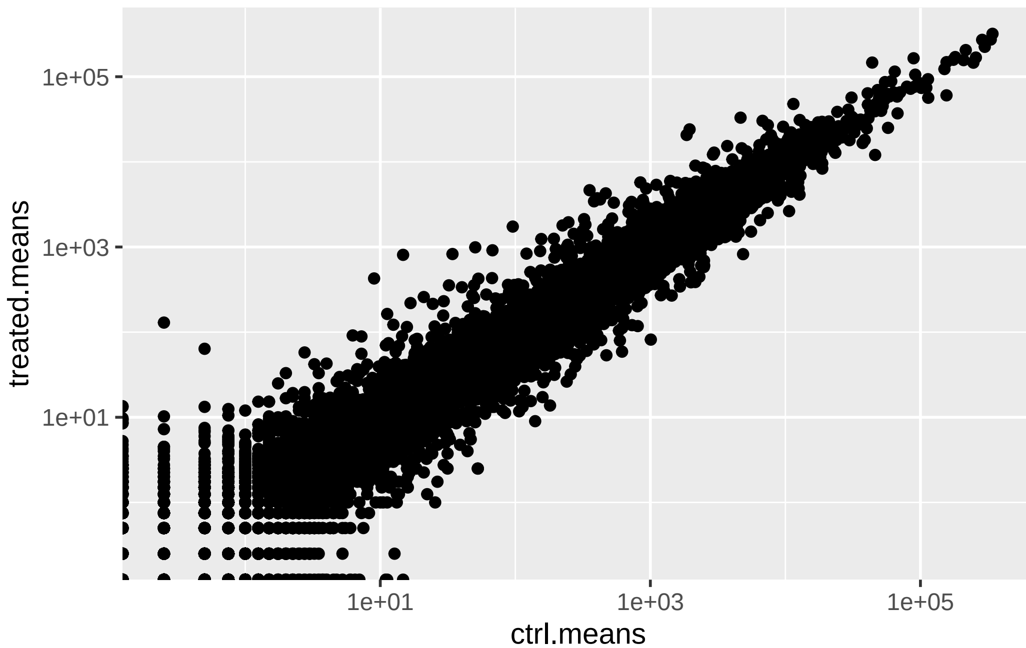
```
Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted
from logarithmic plot
```




```r
library(ggplot2)
ggplot(meancounts, aes(x=ctrl.means, y=treated.means)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10()
```

```
Warning in scale_x_log10(): log-10 transformation introduced infinite values.

Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```



We use log 2 "fold-change" as a way to compare

```
meancounts$log2fc <- log2(meancounts$treated.means / meancounts$ctrl.means)
zero.vals <- which(meancounts[,1:2]==0, arr.ind=TRUE)
to.rm <- unique(zero.vals[,1])
mycounts <- meancounts[-to.rm,]
head(mycounts)
```

```
                ctrl.means treated.means      log2fc
ENSG00000000003     900.75        658.00 -0.45303916
ENSG00000000419     520.50        546.00  0.06900279
ENSG00000000457     339.75        316.50 -0.10226805
ENSG00000000460      97.25         78.75 -0.30441833
ENSG00000000971    5219.00       6687.50  0.35769358
ENSG00000001036    2327.00       1785.75 -0.38194109
```

A common "rule of thumb" thresshold for calling something "up-regulated" is a log2-fold-change of +2 or greater. For "down-regulated" it is -2 or less.

> Q. How many genes are up-regulated?

```r
sum(mycounts$log2fc >= 2)
```

```
[1] 314
```

> Q. How many genes are down-regulated?

```r
sum(mycounts$log2fc <= -2)
```

```
[1] 485
```

```r
head(which(meancounts[, 1:2] == 0, arr.in=T))
```

```
                row col
ENSG00000000005   2   1
ENSG00000004848  65   1
ENSG00000004948  70   1
ENSG00000005001  73   1
ENSG00000006059 121   1
ENSG00000006071 123   1
```

```r
zero.inds <- which(meancounts[,1:2] == 0, arr.ind=TRUE)[,1] #get the row info only
mygenes <- meancounts[-zero.inds, ] #remove those rows with 0 values
```

```r
sum(mygenes$log2fc >= 2)
```

```
[1] 314
```

DESeq2 Analysis

Let's do this with DESeq2 and put some stats behind these numbers.

```r
library(DESeq2)
```

DESeq wants 3 things for analysis, countData, ColData, and design.

```r
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 is the DESeq package to run analysis is called `DESeq()`

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

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

Get the results out of this DESeq object with the function `results()`

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

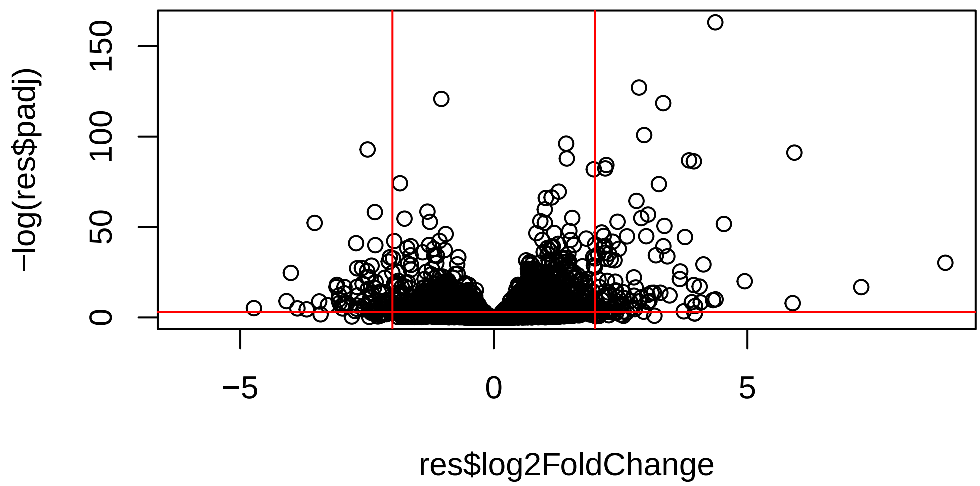
```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 6 columns
                 baseMean log2FoldChange     lfcSE      stat    pvalue
                <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003 747.194195      -0.350703  0.168242 -2.084514 0.0371134
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160       0.206107  0.101042  2.039828 0.0413675
```

```
ENSG00000000457 322.664844       0.024527  0.145134  0.168996 0.8658000
ENSG00000000460  87.682625      -0.147143  0.256995 -0.572550 0.5669497
ENSG00000000938   0.319167      -1.732289  3.493601 -0.495846 0.6200029
                      padj
                 <numeric>
ENSG00000000003  0.163017
ENSG00000000005        NA
ENSG00000000419  0.175937
ENSG00000000457  0.961682
ENSG00000000460  0.815805
ENSG00000000938        NA
```

Volcano Plot

This is a plot of log2FC vs adjusted p-value

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

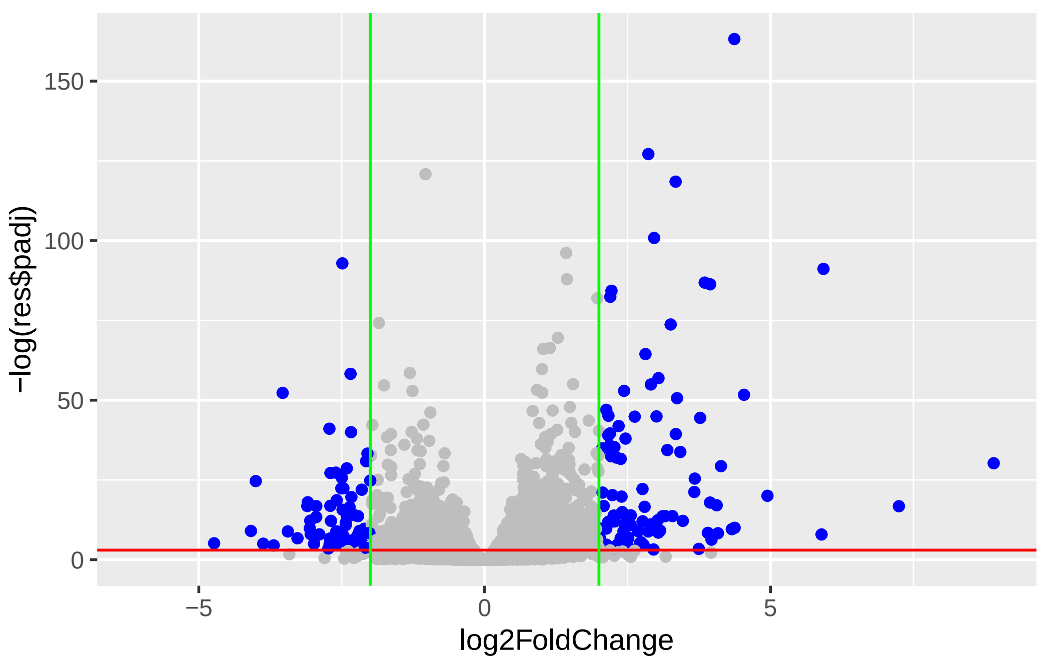


A nicer ggplot volcano plot

```r
mycols <- rep("gray", nrow(res))
mycols[abs(res$log2FoldChange) > 2] <- "blue"
mycols[res$padj >= 0.05] <- "gray"

ggplot(res, aes(log2FoldChange, -log(res$padj))) +
  geom_point(col = mycols) +
  geom_hline(yintercept = -log(0.05), col = "red")+
  geom_vline(xintercept = c(-2,2), col = "green")
```

```
Warning: Removed 23549 rows containing missing values or values outside the scale range
(`geom_point()`).
```



Save our Results

```r
write.csv(res, file="myresults.csv")
```