Class 13 | RNASeq Analysis

AUTHOR
Christopher Leone | A16731724

Background

Today we will analyze some RNA sequencing data on the effects of a common steroid drug on airway cell lines.

There are two main inputs we need for this analysis:

- `countData` : counts for genes in rows with experiments in the columns
- `colData` : the metadatathat tells us about the design of the experiment.

```
# Let's (1) load the libraries:
library(BiocManager)
library(DESeq2)

# And (2) import the files:
counts <- read.csv("airway_scaledcounts.csv", row.names=1)
metadata <-  read.csv("airway_metadata.csv")
```

> (Q1): How many genes are in this dataset?

```
nrow(counts)
```

```
[1] 38694
```

There are **38,694 genes** in this dataset.

> (Q2): How many control cell lines do we have?

```
table(metadata$dex)
```

```
control treated
      4       4
```

There are 4 control cell lines in this table.

Toy Differential Gene Expression

Let's try finding the average or mean of the "control" and "treated" columns and see if they differ.

- First, we need to find al "control" columns
- We need to extract just those columns
- Calculate the `mean()` for each gene "control" values4

```
# I like the dplyr system, so I will use it here:
library(dplyr)
```

```
# (1) Filtering for "control" rows only:
control <- metadata %>% filter(dex=="control")
control.counts <- counts %>% select(control$id)

# (2) Taking and storing the means, displaying the head
control.mean <- rowSums(control.counts)/4
head(control.mean)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457
ENSG00000000460
         900.75            0.00          520.50          339.75
97.25
ENSG00000000938
           0.75
```

> (Q3): Do the same for "treated: to get a `treated.mean`.

```
# (1) Filtering for "treated" rows only:
treated <- metadata %>% filter(dex=="treated")
treated.counts <- counts %>% select(treated$id)

# (2) Taking and storing the means, displaying the head
treated.mean <- rowSums(treated.counts)/4
head(treated.mean)
```

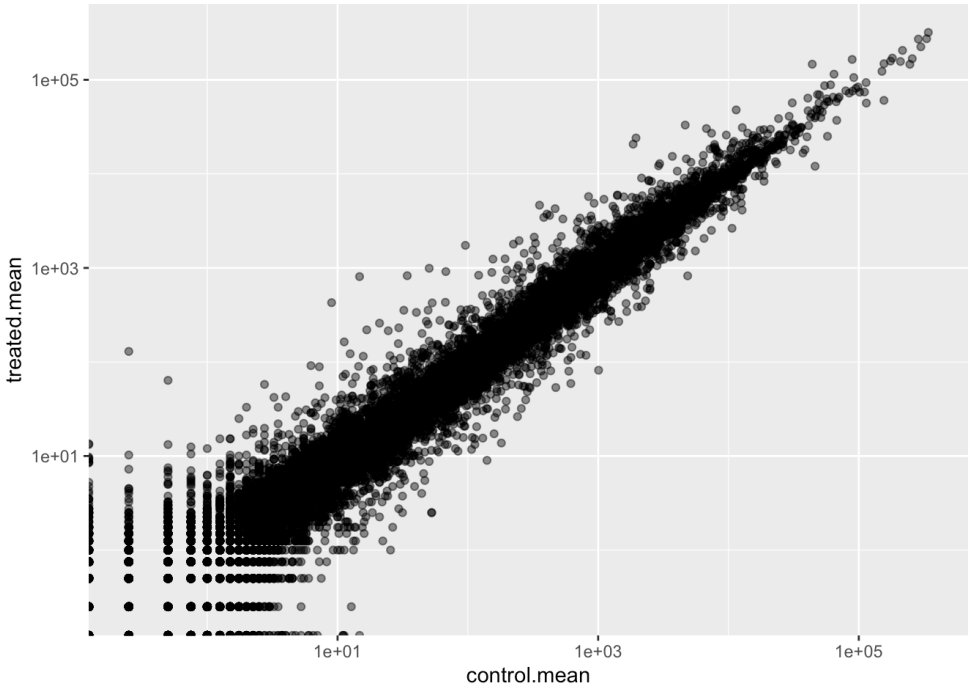
```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457
ENSG00000000460
         658.00            0.00          546.00          316.50
78.75
ENSG00000000938
           0.00
```

> (Q4): And create a plot of `control.mean` vs `treated.mean`.

Ultimately, I decided to put this on logarithmic axes due to the original plot showing most points overlapping. This gives us a much more useful plot.

```
# Let's load the library and make a DF:
library(ggplot2)
means <- data.frame(control.mean, treated.mean)

# Then make the plot:
ggplot(means) +
  aes(x=control.mean, y=treated.mean) +
  geom_point(alpha=0.5) +
  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.
```



A common “rule-of-thumb” is to focus on genes with a log2 “fold-change” of +/-2. This would indicate a significant up/down regulation.

What if we wanted our axes on a `log2()` scale? Let’s change our previous plot:

```
# We will add a log2 fold-change to a table and make a base plo
means$log2 <- log2(means$treated.mean/means$control.mean)
```

> (Q5): Remove any "zero count" genes from our dataset for further analysis

We end up with 21817 genes that do not have any zero values.

```
# We have to omit anything with zero values
to.keep <- rowSums(means[,1:2] == 0) == 0
mycounts <- means[to.keep,]
head(mycounts)
```

	control.mean	treated.mean	log2
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

> (Q6): How many genes are upregulated? What about downregulated?

There are 314 upregulated genes (according to our parameters), and 485 downregulated genes.

```
sum(mycounts$log2 >= 2)
```

```
[1] 314
```

```
sum(mycounts$log2 <= -2)
```

```
[1] 485
```

DESeq2 Analysis

Let's do this properly and consdier the stats—are the differences in the means significant? Let's use `DESeq2` for this.

The first function we will use from this package sets up the input in the particular format that DESeq wants.

```
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
```

We can now run our DESeq analysis:

```
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)
```

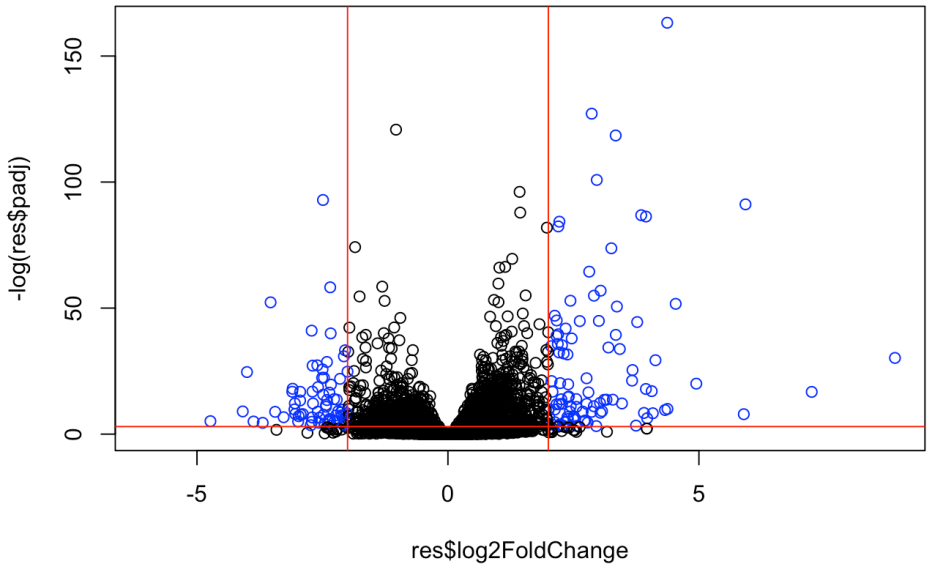
```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 6 columns
                  baseMean log2FoldChange     lfcSE      stat
pvalue
                 <numeric>      <numeric> <numeric> <numeric>
<numeric>
ENSG00000000003 747.194195     -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
```

Result Figure: Volcano Plots

Here, we will be plotting the adjusted P-values (`padj`) vs the `log2fc` . We are looking for very small P-values.

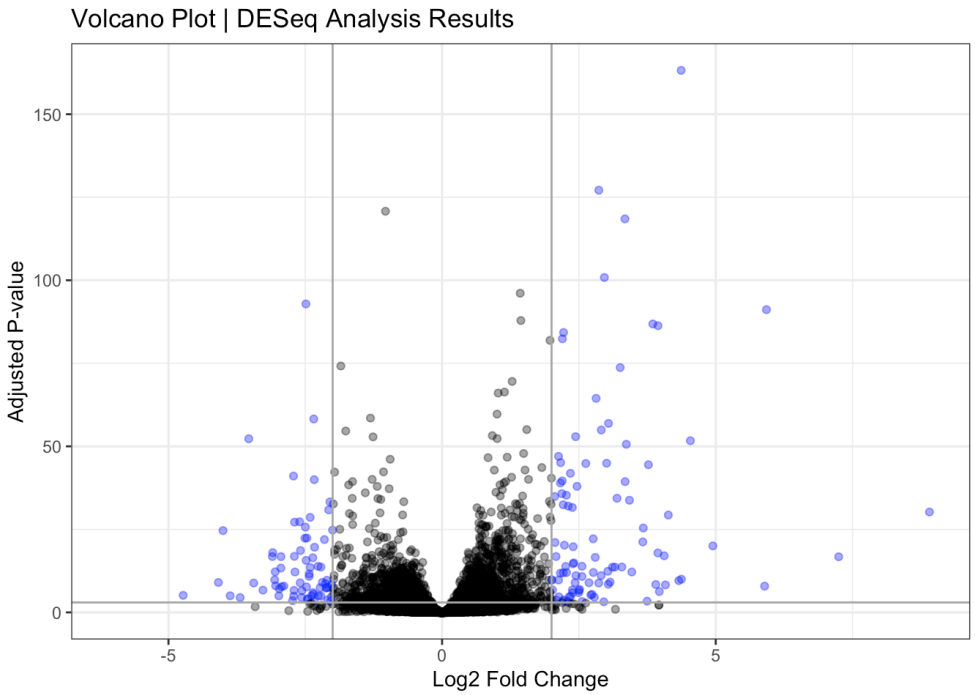
```
# To color the points:
mycols = rep("black", nrow(res))
mycols[res$log2FoldChange <= -2] <- "blue"
mycols[res$log2FoldChange >= 2] <- "blue"
mycols[res$padj >= 0.05] <- "black"

# The plot:
plot(res$log2FoldChange, -log(res$padj), col=mycols)

# Finally, the boundaries of significant stats:
abline(v=2, col="red")
abline(v=-2, col="red")
abline(h=-log(0.05), col="red")
```



Let's do better with a `ggplot()`:

```
ggplot(as.data.frame(res)) +
  aes(x=log2FoldChange, y=-log(padj)) +
  geom_point(alpha=0.4, col=mycols) +
  geom_vline(xintercept = c(-2, 2), col="darkgray") +
  geom_hline(yintercept = -log(0.05), col="darkgray") +
  theme_bw() +
  labs(title = "Volcano Plot | DESeq Analysis Results",
       x="Log2 Fold Change",
       y="Adjusted P-value")
```

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


Volcano Plot | DESeq Analysis Results
Adjusted P-value
150
100
50
0
-5
0
5
Log2 Fold Change