

# Class 13: Transcriptomics and the analysis of RNA-Seq data

Kate Zhou (PID: A17373286)

## Table of contents

Background . . . . .	1
Toy Different Gene Expression . . . . .	2
<b>DESeq2 analysis</b>	<b>7</b>
<b>Result figure: Volcano Plots</b>	<b>9</b>
We need to add gene annotation . . . . .	13
Pathway Analysis . . . . .	15

## 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 experieents in the columns
- `colData`: or metadata that tells us about the design of the experiment (i.e what is in the columns of `countData`)

```
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
```

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

```
nrow(metadata[metadata$dex == 'control',])
```

```
[1] 4
```

```
# or  
table(metadata$dex)
```

```
control treated  
4 4
```

## Toy Different 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 all “control” columns
- Extract just the “control” values for each gene
- Calculate the `mean()` for each gene “control” values

The \$dex column tells me whether we have “control” or “treated”

```
control.id <- metadata[metadata$dex == "control", 1]  
control.counts <- counts[, control.id]
```

Calculate the mean value for each gene in these “control” columns

```
control.mean <- rowMeans(control.counts)
```

Q3. Do the same for “treated” to get a `treated.mean`

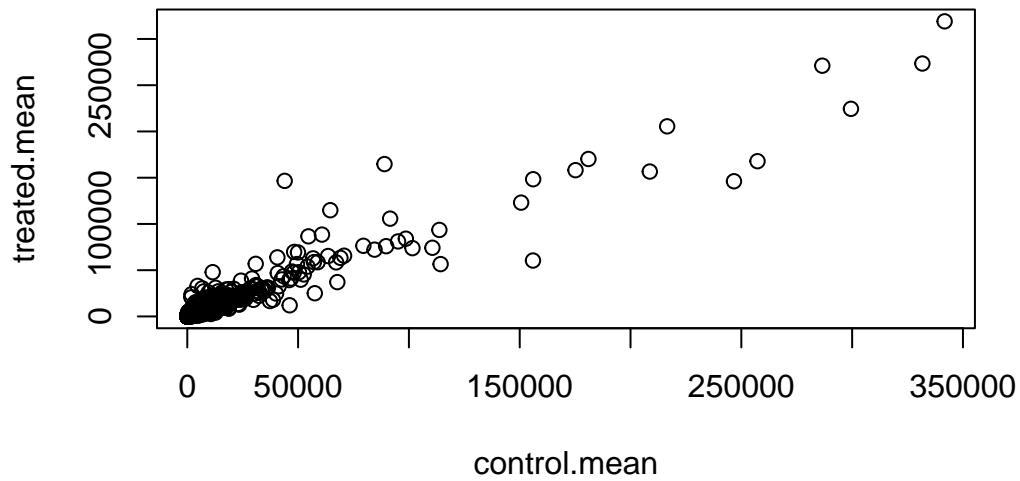
```
treated.id <- metadata[metadata$dex == "treated", 1]  
treated.counts <- counts[, treated.id]  
treated.mean <- rowMeans(treated.counts)
```

Q4. Make a plot of `control.mean` vs `treated.mean`

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

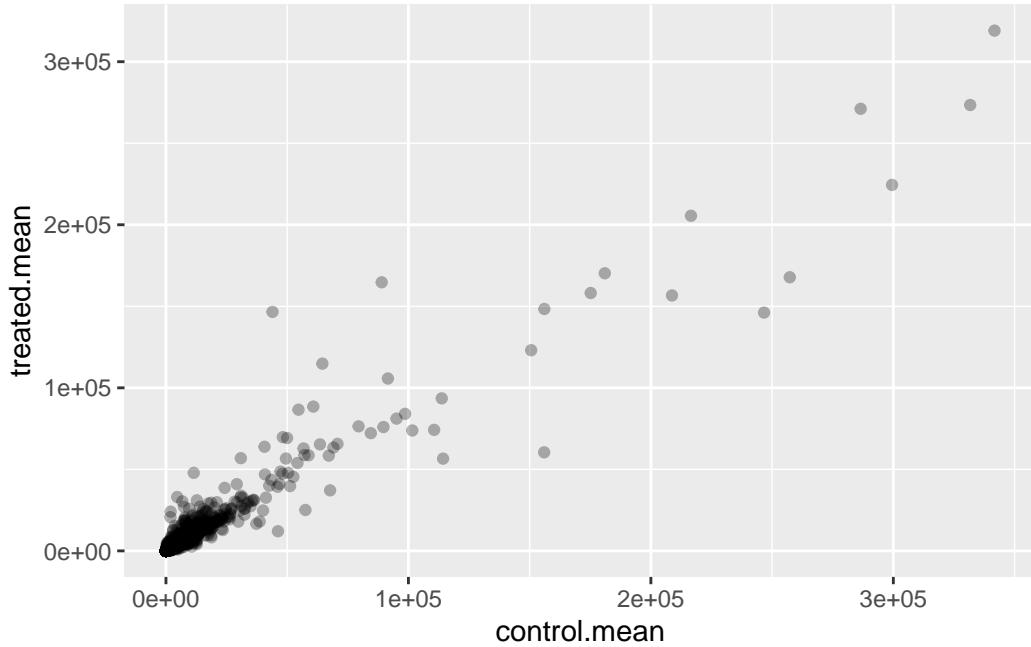
	control.mean	treated.mean
ENSG000000000003	900.75	658.00
ENSG000000000005	0.00	0.00
ENSG00000000419	520.50	546.00
ENSG00000000457	339.75	316.50
ENSG00000000460	97.25	78.75
ENSG00000000938	0.75	0.00

```
plot(meancounts)
```



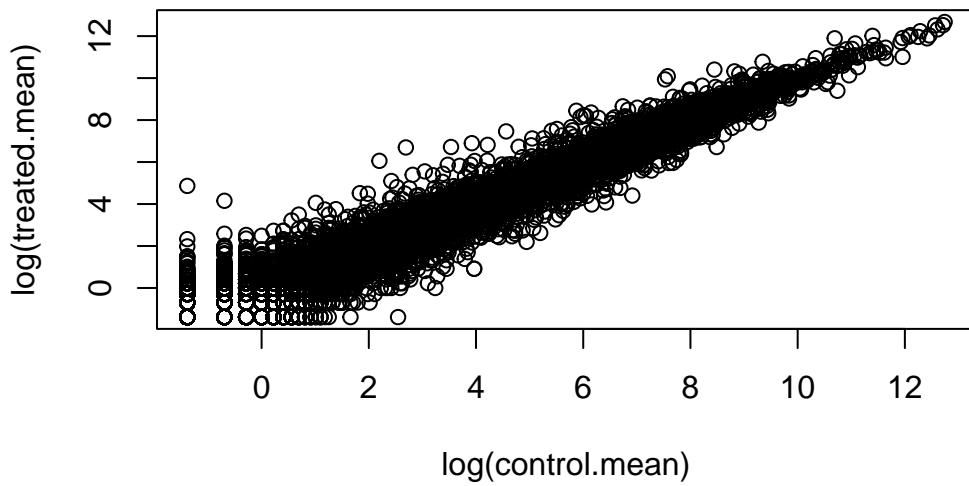
```
library(ggplot2)

ggplot(meancounts) +
  aes(control.mean, treated.mean) +
  geom_point(alpha = 0.3)
```



We totally need to log transform this data as it is so heavily skewed

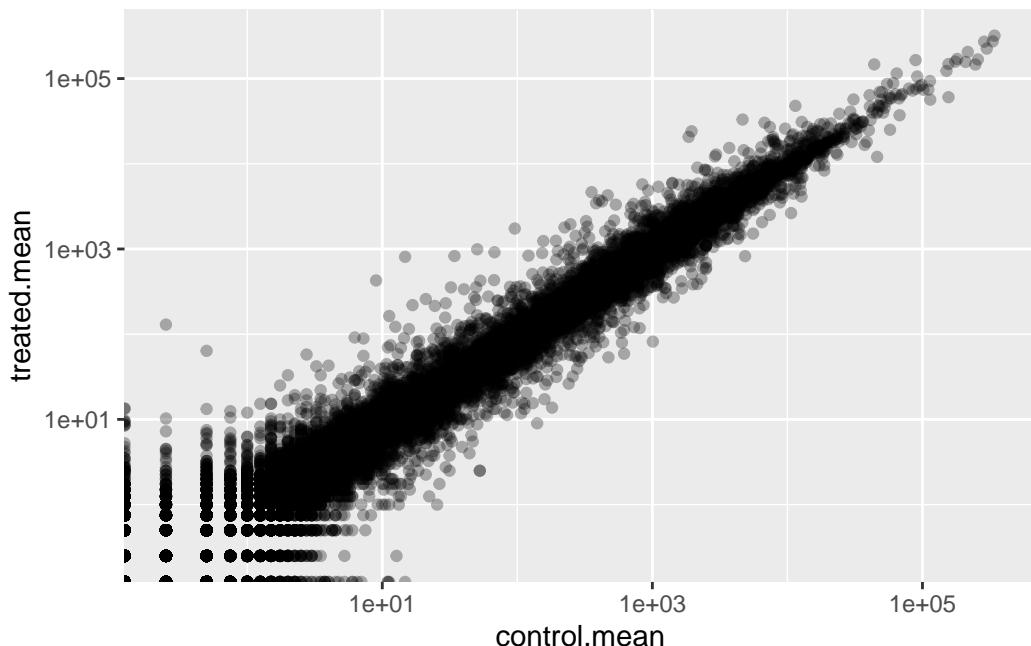
```
plot(log(control.mean), log(treated.mean))
```



```
ggplot(meancounts) +  
  aes(control.mean, treated.mean) +  
  geom_point(alpha = 0.3) +  
  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.



```
# Treated/control  
  
log2(20/20)
```

[1] 0

Doubling of the amount

```
log2(40/20)
```

```
[1] 1
```

Half of the amount

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

```
[1] -1
```

A common “rule-of-thumb” is to focus on genes with a log2 “fold-change” of +2 as so-called UP REGULATED and -2 DOWN REGULATED

```
log2(80/20)
```

```
[1] 2
```

Let's add a log2 fold-change value to our `meancounts` data.frame

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

head(meancounts)
```

	control.mean	treated.mean	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

Q5. Remove any “zero count” genes from our dataset for further analysis

```
to.rm <- rowSums(meancounts[, 1:2] == 0) > 0
to.keep <- rowSums(meancounts[, 1:2] == 0) == 0
sum(to.keep)
```

```
[1] 21817
```

```
mycounts <- meancounts[to.keep,]  
head(mycounts)
```

	control.mean	treated.mean	log2fc
ENSG000000000003	900.75	658.00	-0.45303916
ENSG000000000419	520.50	546.00	0.06900279
ENSG000000000457	339.75	316.50	-0.10226805
ENSG000000000460	97.25	78.75	-0.30441833
ENSG000000000971	5219.00	6687.50	0.35769358
ENSG00000001036	2327.00	1785.75	-0.38194109

Q6. How many genes are “up” regulated at a log2fc threshold of +2?

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

[1] 314

Q7. How many genes are “down” regulated at a log2fc threeshold of -2?

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

[1] 485

## DESeq2 analysis

Let’s do this properly and consider the stats - are the differences in the means significant?

We will use DESeq2 to do this:

```
library(DESeq2)
```

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)
```

Peak at results

```
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

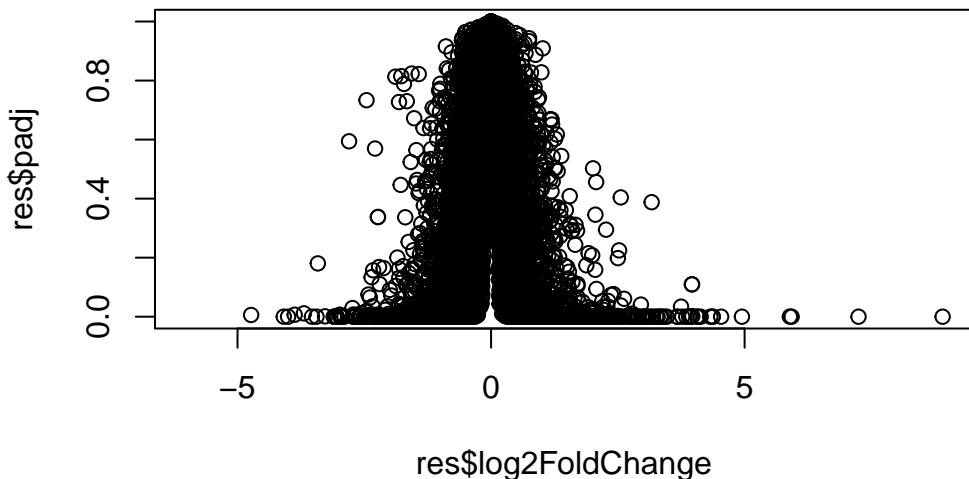
```
padj
```

```
<numeric>
ENSG000000000003 0.163035
ENSG000000000005 NA
ENSG000000000419 0.176032
ENSG000000000457 0.961694
ENSG000000000460 0.815849
ENSG000000000938 NA
```

## Result figure: Volcano Plots

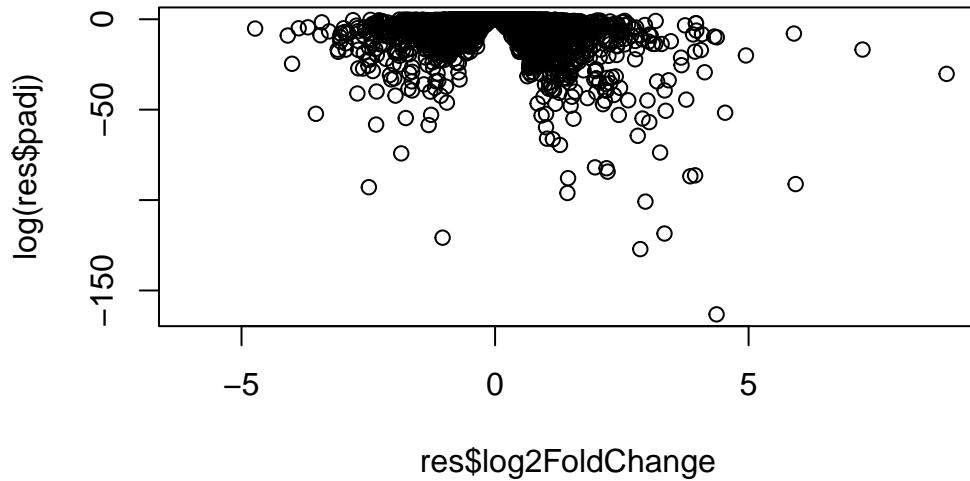
Plot of the P-value cs the log2FC

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



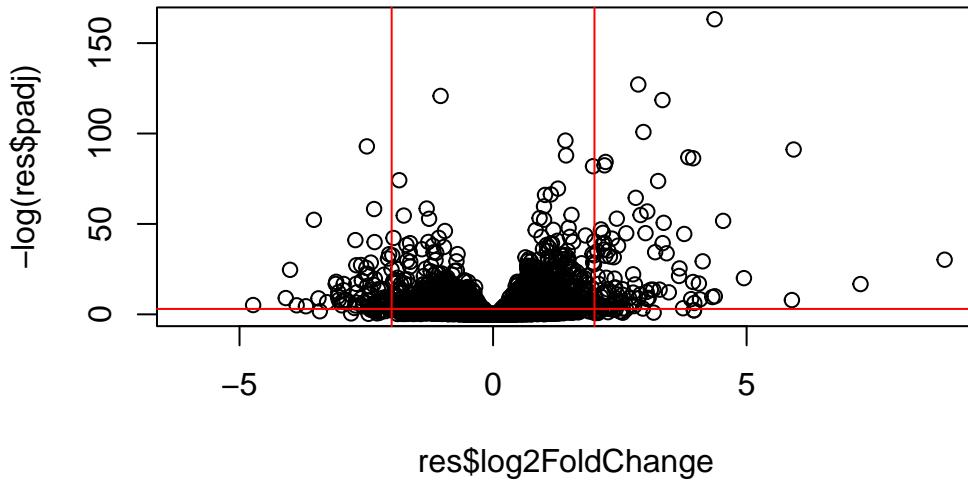
This P-value data is again heavily skewed so lets log transform it

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



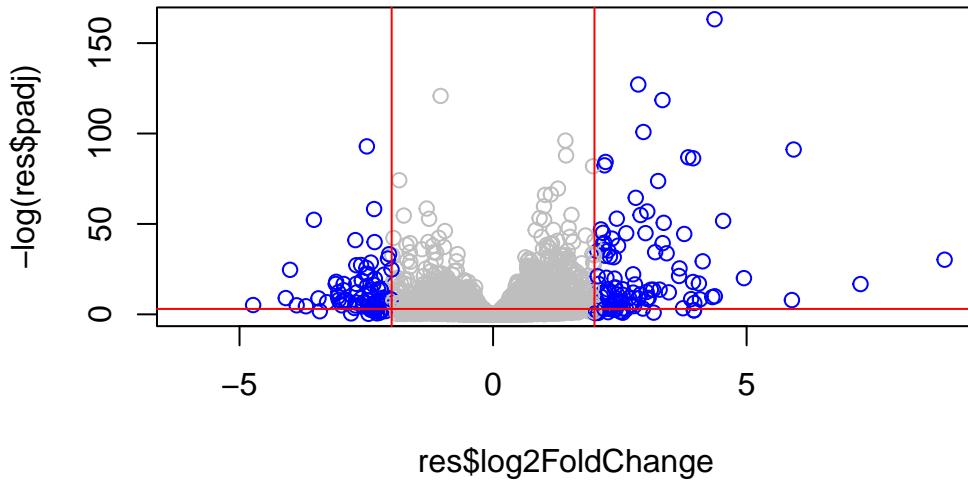
We can flip the y-axis by adding a minus sign

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



Let's add some color.

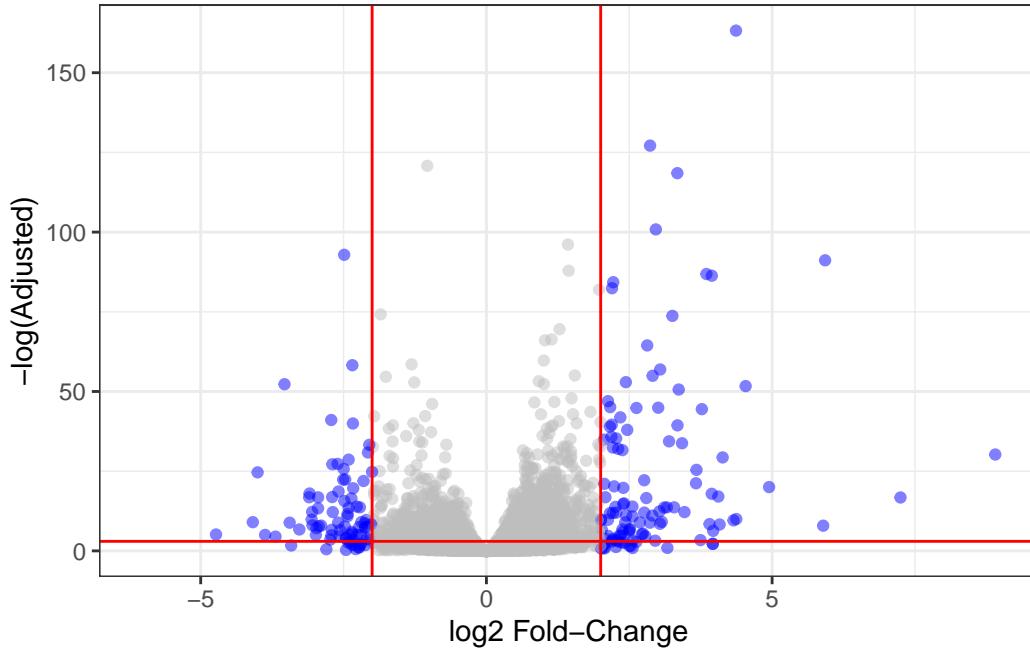
```
mycols <- rep("grey", nrow(res))
mycols[res$log2FoldChange <= -2] <- "blue"
mycols[res$log2FoldChange >= 2] <- "blue"
plot(res$log2FoldChange, -log(res$padj), col=mycols)
abline(v=-2, col= "red")
abline(v=2, col= "red")
abline(h=-log(0.05), col="red")
```



Q. Make a ggplot volcano plot with colors and lines as annotation along with nice axis labels.

```
ggplot(as.data.frame(res)) +
  aes(res$log2FoldChange, -log(res$padj)) +
  geom_point(alpha = 0.5, col=mycols) +
  geom_vline(xintercept = c(-2, 2), col = "red") +
  geom_hline(yintercept = -log(0.05), col = "red") +
  theme_bw() +
  labs(x="log2 Fold-Change",
       y="-log(Adjusted)")
```

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



## We need to add gene annotation

Gene symbols and different database IDs

```
head(rownames(res))
```

```
[1] "ENSG00000000003" "ENSG00000000005" "ENSG00000000419" "ENSG00000000457"
[5] "ENSG00000000460" "ENSG00000000938"
```

Install from bioconductor with `BiocManager::install("AnnotationDbi")`

```
library("AnnotationDbi")
library("org.Hs.eg.db")
```

What different database ID types can I translate between.

```
columns(org.Hs.eg.db)
```

```
[1] "ACNUM"      "ALIAS"       "ENSEMBL"      "ENSEMLPROT"   "ENSEMLTRANS"
[6] "ENTREZID"   "ENZYME"      "EVIDENCE"     "EVIDENCEALL"  "GENENAME"
[11] "GENETYPE"   "GO"          "GOALL"        "IPI"          "MAP"
[16] "OMIM"        "ONTOLOGY"    "ONTOLOGYALL" "PATH"         "PFAM"
[21] "PMID"        "PROSITE"     "REFSEQ"       "SYMBOL"      "UCSCKG"
[26] "UNIPROT"
```

Let's "map" between "ENSEMBL" and "SYMBOL" (i.e. gene symbol)

Add

```
res$symbol <- mapIds(x=org.Hs.eg.db,
                      keys=rownames(res),
                      keytype="ENSEMBL",
                      column="SYMBOL")
```

'select()' returned 1:many mapping between keys and columns

```
res$name <- mapIds(x=org.Hs.eg.db,
                     keys=rownames(res),
                     keytype="ENSEMBL",
                     column="GENENAME")
```

'select()' returned 1:many mapping between keys and columns

```
res$entrez <- mapIds(x=org.Hs.eg.db,
                      keys=rownames(res),
                      keytype="ENSEMBL",
                      column="ENTREZID")
```

'select()' returned 1:many mapping between keys and columns

```
head(res)
```

```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 9 columns
  baseMean log2FoldChange      lfcSE      stat     pvalue
  <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003  747.194195     -0.3507030  0.168246 -2.084470  0.0371175
```

ENSG000000000005	0.000000	NA	NA	NA	NA
ENSG000000000419	520.134160	0.2061078	0.101059	2.039475	0.0414026
ENSG000000000457	322.664844	0.0245269	0.145145	0.168982	0.8658106
ENSG000000000460	87.682625	-0.1471420	0.257007	-0.572521	0.5669691
ENSG000000000938	0.319167	-1.7322890	3.493601	-0.495846	0.6200029
	padj	symbol		name	entrez
	<numeric>	<character>		<character>	<character>
ENSG000000000003	0.163035	TSPAN6	tetraspanin 6		7105
ENSG000000000005	NA	TNMD	tenomodulin		64102
ENSG000000000419	0.176032	DPM1	dolichyl-phosphate m..		8813
ENSG000000000457	0.961694	SCYL3	SCY1 like pseudokina..		57147
ENSG000000000460	0.815849	FIRRM	FIGNL1 interacting r..		55732
ENSG000000000938	NA	FGR	FGR proto-oncogene, ..		2268

Be sure to save our annotated results to a file.

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

## Pathway Analysis

We first need to add gene symbols (e.g. HBB etc). So we know what genes we are dealing with. We need to “translate” between ENSEMBLE ids that we have in

Install the packages we need for pathway analysis: Run in your R console (i.e. not your Quarto doc!) `BiocManager::install( c("pathview", "gage", "gageData") )`

```
library(gage)
library(gageData)
library(pathview)
```

Let's peak at kegg

```
data("kegg.sets.hs")
```

```
head(kegg.sets.hs, 2)
```

```
$`hsa00232 Caffeine metabolism`
[1] "10"    "1544"   "1548"   "1549"   "1553"   "7498"   "9"

$`hsa00983 Drug metabolism - other enzymes`
[1] "10"      "1066"    "10720"   "10941"   "151531"  "1548"    "1549"    "1551"
```

```
[9] "1553"   "1576"   "1577"   "1806"   "1807"   "1890"   "221223" "2990"
[17] "3251"   "3614"   "3615"   "3704"   "51733"  "54490"  "54575"   "54576"
[25] "54577"  "54578"  "54579"  "54600"  "54657"  "54658"  "54659"   "54963"
[33] "574537" "64816"  "7083"   "7084"   "7172"   "7363"   "7364"   "7365"
[41] "7366"   "7367"   "7371"   "7372"   "7378"   "7498"   "79799"  "83549"
[49] "8824"   "8833"   "9"      "978"
```

To run pathway analysis we will use the `gage()` function and it requires a wee “vector of importance”. We will use our Log2FC results from our `res` object.

```
foldchanges = res$log2FoldChange
names(foldchanges) = res$entrez
head(foldchanges)
```

```
7105          64102          8813          57147          55732          2268
-0.35070302           NA  0.20610777  0.02452695 -0.14714205 -1.73228897
```

```
keggres = gage(foldchanges, gsets=kegg.sets.hs)
attributes(keggres)
```

```
$names
[1] "greater" "less"     "stats"
```

```
head(keggres$less)
```

	p.geomean	stat.mean	p.val	q.val	set.size	exp1
hsa05332 Graft-versus-host disease	0.0004250461	-3.473346	0.0004250461	0.09053483		
hsa04940 Type I diabetes mellitus	0.0017820293	-3.002352	0.0017820293	0.14232581		
hsa05310 Asthma	0.0020045888	-3.009050	0.0020045888	0.14232581		
hsa04672 Intestinal immune network for IgA production	0.0060434515	-2.560547	0.0060434515	0.31387180		
hsa05330 Allograft rejection	0.0073678825	-2.501419	0.0073678825	0.31387180		
hsa04340 Hedgehog signaling pathway	0.0133239547	-2.248547	0.0133239547	0.47300039		

hsa05332 Graft-versus-host disease	40	0.0004250461
hsa04940 Type I diabetes mellitus	42	0.0017820293
hsa05310 Asthma	29	0.0020045888
hsa04672 Intestinal immune network for IgA production	47	0.0060434515
hsa05330 Allograft rejection	36	0.0073678825
hsa04340 Hedgehog signaling pathway	56	0.0133239547

We can pass our foldchanges vector (our results) together with

```
pathview(gene.data=foldchanges, pathway.id="hsa05310")
```

'select()' returned 1:1 mapping between keys and columns

Info: Working in directory /Users/katez/Desktop/BIMM143/class13

Info: Writing image file hsa05310.pathview.png

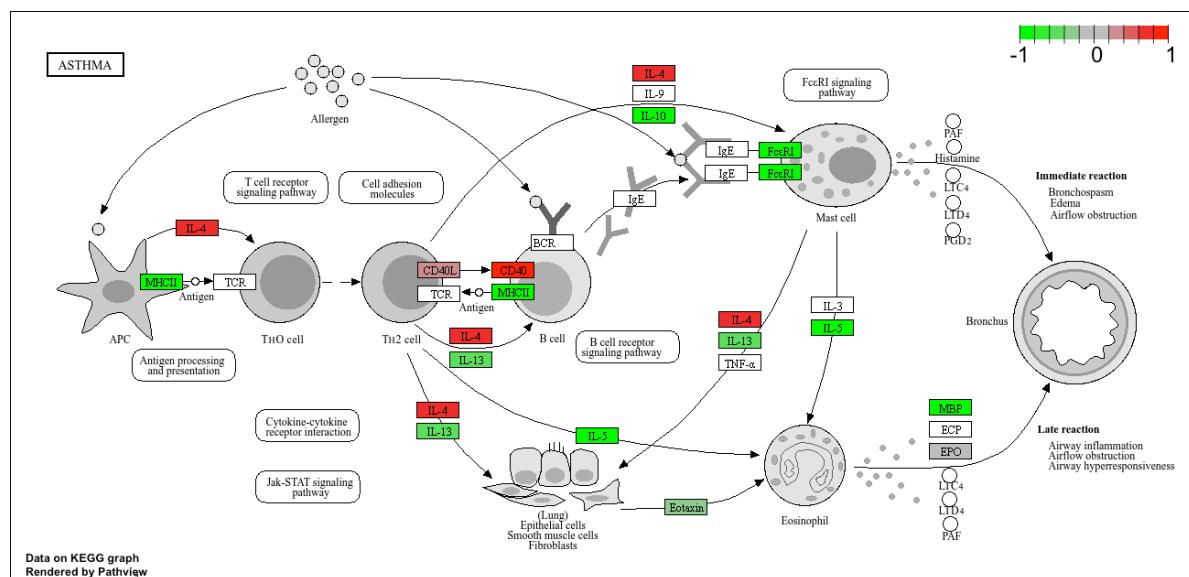


Figure 1: The Asthma pathway overlaps with our differentially expressed genes