BIMM 143 Lab 13

Sofia Lanaspa, A17105313

Today we will analyze some RNASeq data from Himes et al. on the effets of dexamethasone (dex), a syntehtic glucorticoid steroid on airway smooth muscle cells (ASM)!

Download BiocManager in R console

```
# BiocManager::install("DESeq2")
```

Check installations were executed correctly

```
suppressMessages(library(BiocManager))
suppressMessages(library(DESeq2))
# suppress message is to hide output messages of download
```

Read data for today and save to variables

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

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
```

```
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
```

Question 1

Q1. How many genes are in this dataset?

```
nrow(counts)
```

```
[1] 38694
```

Question 2

Q2. How many 'control' cell lines do we have?

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

```
control treated
      4       4
```

Toy differential expression analysis

Calculate the mean per gene count values for all 'control' samples (ex.columns in 'counts') and 'treated' so we can compare them and see effect of the drug on patients

Question 3

Q3. How would you make the above code in either approach more robust? Is there a function that could help here?

1. Find all 'control' values/columns in 'counts'

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

2. Find the mean per gene across all control columns

```
control.mean <- apply(control.counts, 1, mean)
# Alternative way to calculate it:
# 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
```

Question 4

Q4. Follow the same procedure for the treated samples (i.e. calculate the mean per gene across drug treated samples and assign to a labeled vector called treated.mean)

3. Find mean per gene across treated samples

```
treated.inds <- metadata$dex == "treated"
treated.counts <- counts[ ,treated.inds]

treated.mean <- apply(treated.counts, 1, mean)
# Alternative way to calculate it:
# treated.mean <- rowSums(treated.counts) / 4
head(treated.mean)
```

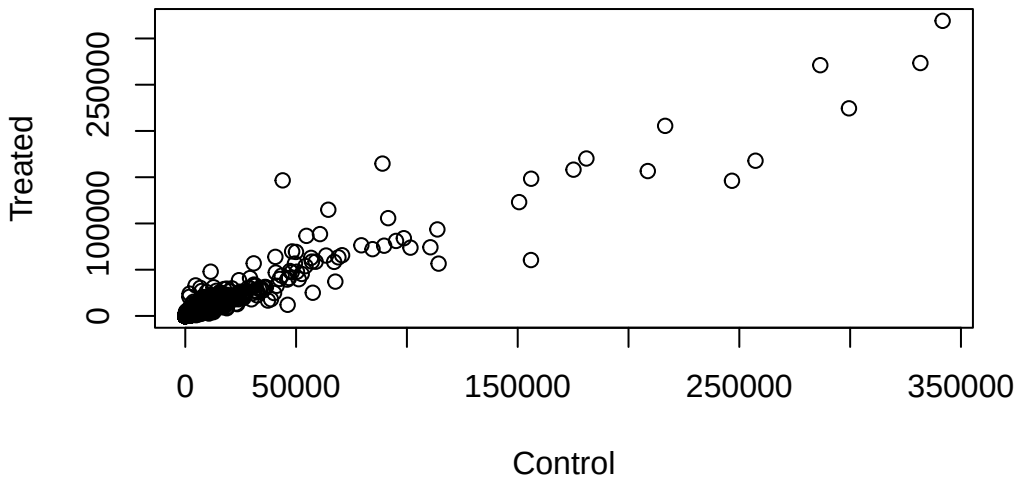
```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
         658.00            0.00          546.00          316.50           78.75
ENSG00000000938
           0.00
```

Question 5a

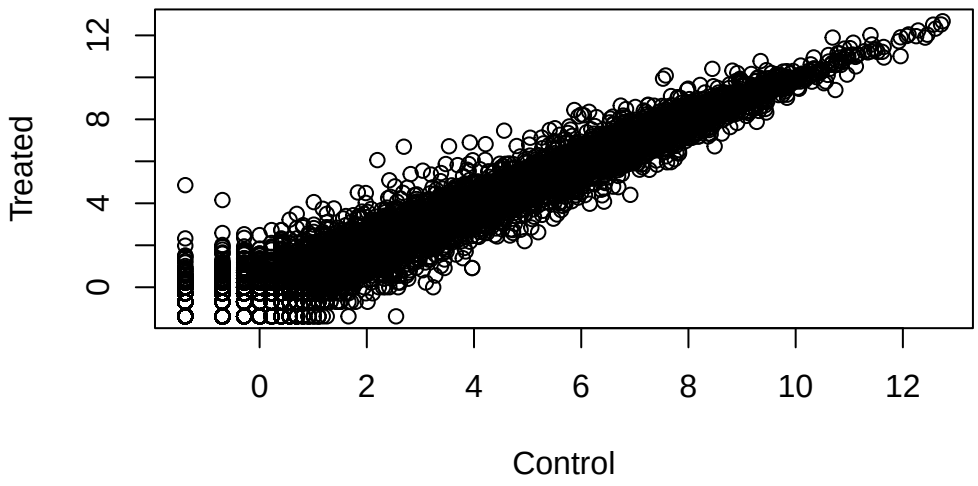
Q5 (a). Create a scatter plot showing the mean of the treated samples against the mean of the control samples. Your plot should look something like the following. Plot control vs treate means:

```
# combine both means into a data frame
meanCounts <- data.frame(control.mean, treated.mean)

# plot values
library(ggplot2)
plot(meanCounts, xlab="Control", ylab="Treated")
```



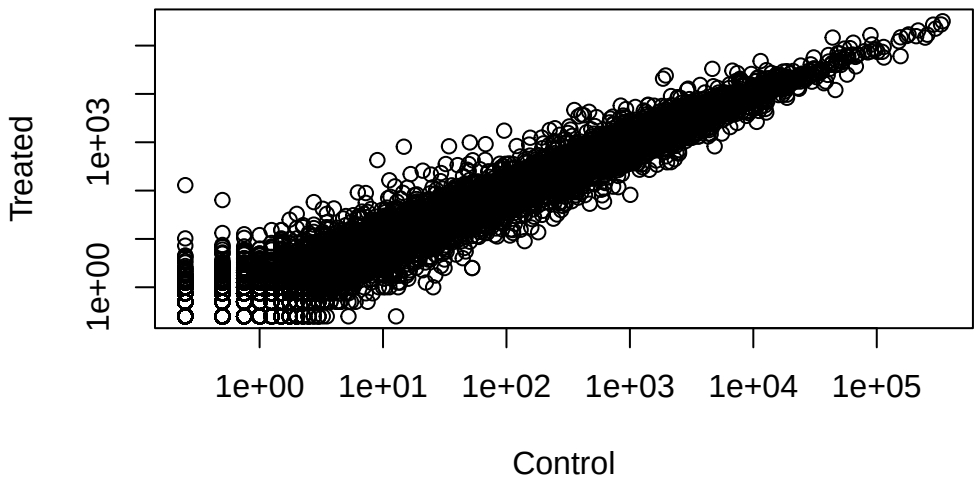

```
# this data looks highly skewed, need to take log to see values more clearly
plot(log(meanCounts), xlab="Control", ylab="Treated")
```




```
plot(meanCounts, log='xy', xlab="Control", ylab="Treated")
```

```
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
```



Most frequently use log2 transformations for this type of data because it makes interpretation of 'fold-change' easier • rule-of-thumb in the filed is a log2 fold-change of +2 or -2 where we start to pay attention:

```
log2(10/10) # get 0 for this, easy to interpret since no difference between sets is represent
```

```
[1] 0
```

```
log2(20/10) # 1 so increase in expression
```

```
[1] 1
```

```
log2(10/20) # -1 so decrease in expression
```

```
[1] -1
```

We see negative values when the treated is lower than control

Let's calculate the log2(fold-change) and add it to our 'meanCounts' data.frame

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

```
                control.mean treated.mean      log2fc
ENSG00000000003       900.75       658.00 -0.45303916
ENSG00000000005         0.00         0.00         NaN
ENSG00000000419       520.50       546.00  0.06900279
ENSG00000000457       339.75       316.50 -0.10226805
ENSG00000000460        97.25        78.75 -0.30441833
ENSG00000000938         0.75         0.00        -Inf
```

```
# give value per gene where if count is 0, both False, 1 is True and False, and 2 is both Tru
# the last > 0 makes sure we only get the genes that we want to remove
to.rm <- rowSums(meanCounts[,1:2] == 0) > 0
myCounts <- meanCounts[!to.rm,]
```

How many genes do I have left after this zero count filtering?

```
nrow(myCounts)
```

```
[1] 21817
```

Q.How many genes are "up" regulated upon drug treatment at a threshold of +2 log2-fold-change?

1. I need to extract the log2fc column
2. I need to find those that are above +2
3. Count them

```
sum(myCounts$log2fc > 2)
```

```
[1] 250
```

Q.How many genes are "down" regulated upon drug treatment at a threshold of +2 log2-fold-change?

```
sum(myCounts$log2fc < -2)
```

```
[1] 367
```

WOWWWW hold on, we are missing stats, is this difference in the mean counts significant?

Let's do this analysis the right way with stats and use the **DESeq** package:

DESeq Analysis

```
# DESeq already loaded at the beginning of document
```

First function we will use will set up the data in the way (format) DESeq wants it

```
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
```

```
dds
```

```
class: DESeqDataSet
dim: 38694 8
metadata(1): version
assays(1): counts
rownames(38694): ENSG00000000003 ENSG00000000005 ... ENSG00000283120
  ENSG00000283123
rowData names(0):
colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521
colData names(4): id dex celltype geo_id
```

The function in the package is called 'DESeq()' and we can run it on our 'dds' object

```
# results(dds) gives error
dds <- DESeq(dds)
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

I will get he results from dds with the 'results()' function:

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

```
out of 25258 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 1563, 6.2%
LFC < 0 (down)     : 1188, 4.7%
outliers [1]       : 142, 0.56%
low counts [2]     : 9971, 39%
(mean count < 10)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

Changing alpha:

```
res05 <- results(dds, alpha=0.05)
summary(res05)
```

```
out of 25258 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 1236, 4.9%
LFC < 0 (down)     : 933, 3.7%
outliers [1]       : 142, 0.56%
low counts [2]     : 9033, 36%
(mean count < 6)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

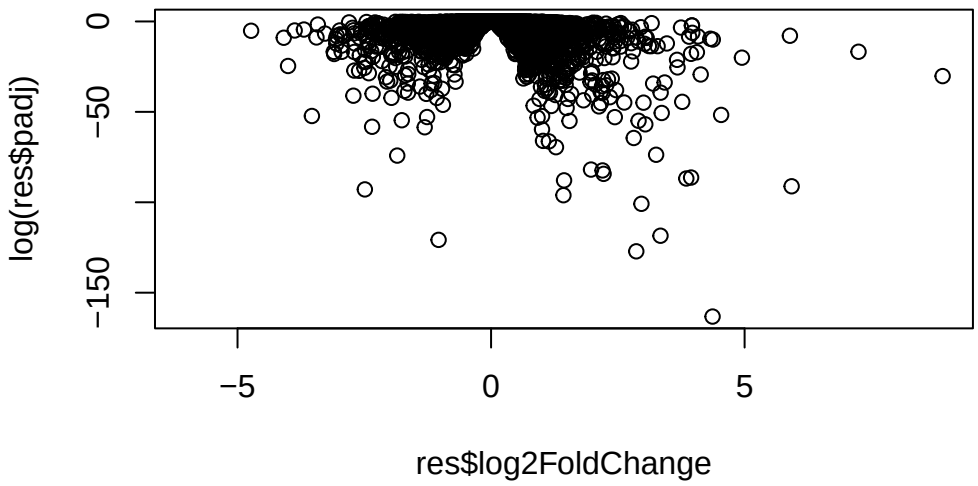
Volcano Plot

Make a common overall results figure from this analysis, called the volcano plot, it is designed to keep our inner biologist and stats nerd happyyy (plots fold-change vs p-value)

** this is doing a stat test of the ~38 thousand genes, but this is an issue because even a small p value of like 0.5% is a lot of values, so need a more strict p value

```
# res$log2FoldChange
# res$padj
```

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




```
# this plot shows lower values have smaller p-values
```

```
log(0.0005)
```

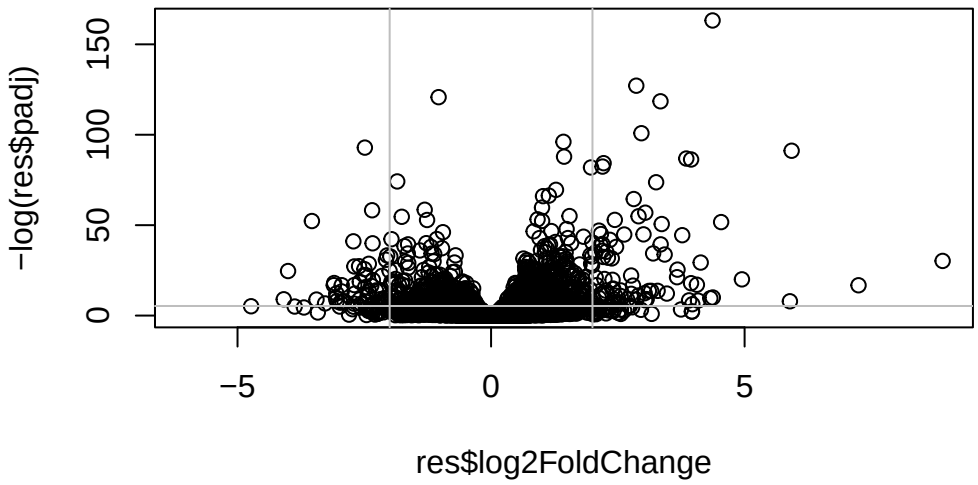
```
[1] -7.600902
```

```
log(0.00000000005)
```

```
[1] -23.719
```

** values on the extremes are good because they indicate a lot of fold-change and very small p-values**

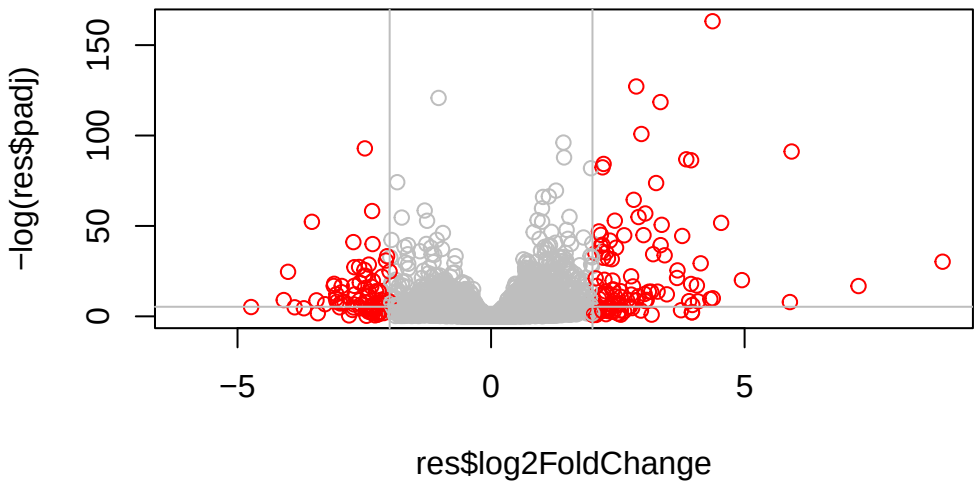
```
plot(res$log2FoldChange, -log(res$padj))
abline(v = c(-2,2), col="gray") # create lines to show quadrant
abline(h = -log(0.005), col="gray") # value in log is chosen alpha to show points we care abo
```



Add color to plot:

```
mycols <- rep("gray",nrow(res))
mycols[abs(res$log2FoldChange) > 2] <- "red"
mycols[res$log2FoldChange < -2] <- "red"
mycols[res$padj < log(0.005)] <- "blue"
# mycols

plot(res$log2FoldChange, -log(res$padj), col = mycols)
abline(v = c(-2,2), col="gray") # create lines to show quadrant
abline(h = -log(0.005), col="gray") # value in log is chosen alpha to show points we care abo
```



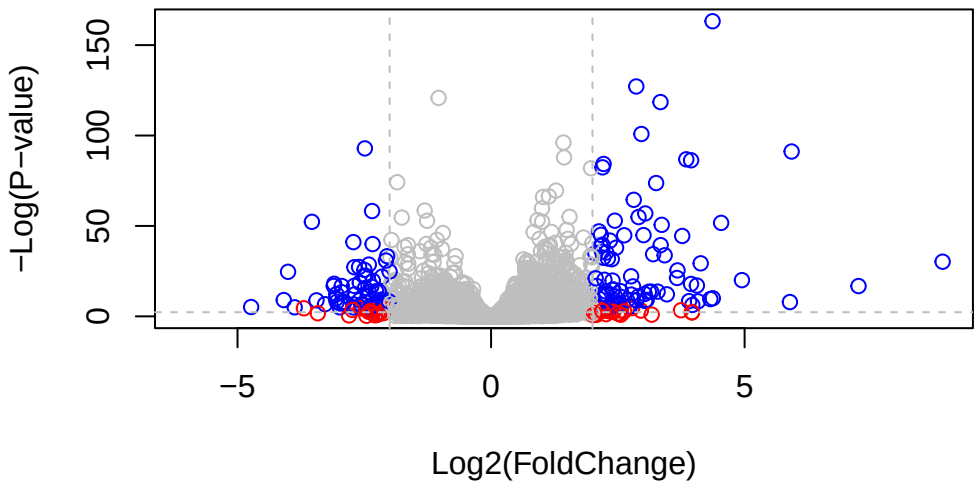
Correctly colored graph

```
# Setup our custom point color vector
mycols <- rep("gray", nrow(res))
mycols[ abs(res$log2FoldChange) > 2 ]  <- "red"

inds <- (res$padj < 0.01) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

# Volcano plot with custom colors
plot(res$log2FoldChange,  -log(res$padj),
 col=mycols, ylab="-Log(P-value)", xlab="Log2(FoldChange)" )

# Cut-off lines
abline(v=c(-2,2), col="gray", lty=2)
abline(h=-log(0.1), col="gray", lty=2)
```



Save my results to date out to disc:

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

We will pick this up next day and add **annotation** (what are these genes of interest) and do **pathway analysis** (what biology) are they known to be involved in! :)

```
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
```

Adding annotation data

I need to translate our gene identifier "ENSGOOOOO..." into names that the rest of the world can understand

To do this "annotation" I will use the **AnnotationDbi** bioconductor package
Install it in the console: BiocManager::install("AnnotationDbi") BiocManager::install("org.Hs.eg.db")

Make sure to do it in the console and type 'n' if called to update

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

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

I will use the 'mapIds()' function to "map" my identifiers to those from different databases. I will o between "ENSEMBL" and "SYMBOL" (and then after gene name)

```
# mapIds(org.Hs.eg.db,
#       keys = rownames(res),
#      keytype = "ENSEMBL",
#     column = "SYMBOL")
```

```
res$symbol <- mapIds(org.Hs.eg.db,
                     keys=row.names(res), # Our genenames
                     keytype="ENSEMBL",        # The format of our genenames
                     column="SYMBOL",          # The new format we want to add
                     multiVals="first")
```

```
'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 7 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      symbol
                <numeric> <character>
ENSG00000000003  0.163035      TSPAN6
ENSG00000000005        NA        TNMD
ENSG00000000419  0.176032        DPM1
ENSG00000000457  0.961694       SCYL3
ENSG00000000460  0.815849       FIRRM
ENSG00000000938        NA         FGR
```

Save our annotated results object

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

Pathway Analysis

Now that we have our results with added annotation we can do some pathway mapping

Let's use the **gage** package to look for KEGG pathways in our results (genes of interest). I will also use the **pathview** package to draw litte pathway figures.

```
# BiocManager::install(c("pathview", "gage", "gageData"))
library(pathview)
```

```
##############################################################################
Pathview is an open source software package distributed under GNU General
Public License version 3 (GPLv3). Details of GPLv3 is available at
```

```
http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to
formally cite the original Pathview paper (not just mention it) in publications
or products. For details, do citation("pathview") within R.

The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG
license agreement (details at http://www.kegg.jp/kegg/legal.html).
##############################################################################
```

```
library(gage)
```

```
library(gageData)

data(kegg.sets.hs)

# Examine the first 2 pathways in this kegg set for humans
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"
```

What **gage** wants as an input is not my big table/data.frame of results, it just wants a "vector of importance". For RNASeq data like we have this is out log2FC values ...

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

```
[1] -0.35070302          NA  0.20610777  0.02452695 -0.14714205 -1.73228897
```

Now lets run the gage pathway analysis

```
# Get the results
keggres = gage(foldchanges, gsets=kegg.sets.hs)
```

What is in this 'keggres' object?

```
attributes(keggres)
```

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

```
head(keggres$less, 3)
```

```
                                      p.geomean stat.mean p.val q.val
hsa00232 Caffeine metabolism                 NA       NaN    NA    NA
hsa00983 Drug metabolism - other enzymes     NA       NaN    NA    NA
hsa01100 Metabolic pathways                  NA       NaN    NA    NA
                                      set.size exp1
hsa00232 Caffeine metabolism                 0   NA
hsa00983 Drug metabolism - other enzymes     0   NA
hsa01100 Metabolic pathways                  0   NA
```

Highlight KEGG pathways with out genes highlighted "hsa05310 Asthma"

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

```
Warning: None of the genes or compounds mapped to the pathway!
Argument gene.idtype or cpd.idtype may be wrong.

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

Info: Working in directory /Users/sofialanaspa/Desktop/BIMM 143 (bioinf)/BIMM 143 Lab 13

Info: Writing image file hsa05310.pathview.png
```

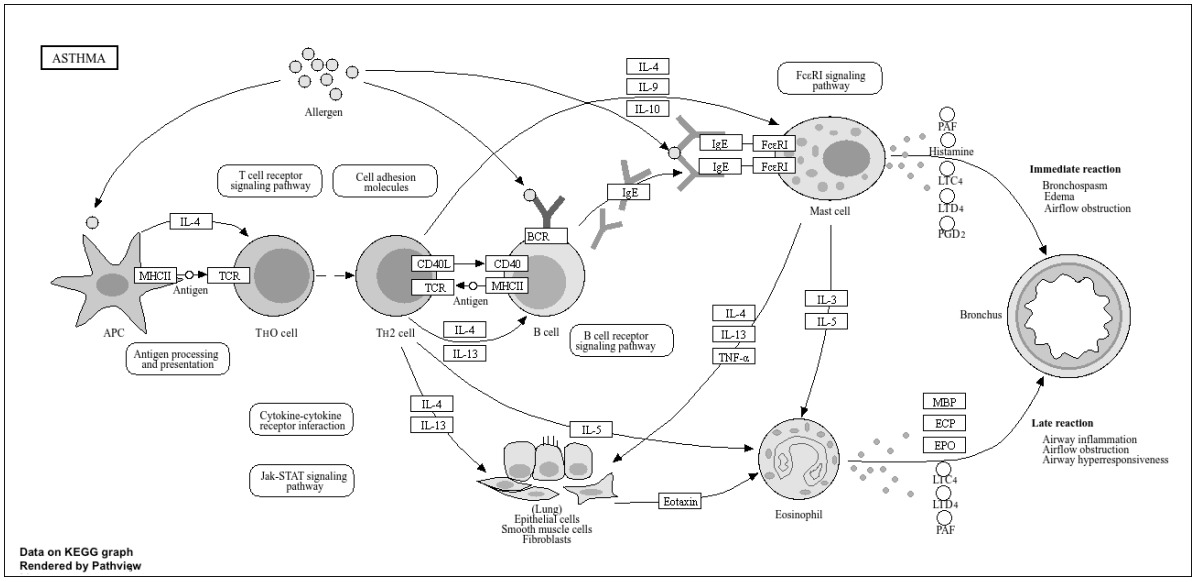


Figure 1: Asthma pathway with my DEGs