Class 13: Transcriptomics, RNA-Seq Analysis

Rahul Nedunuri (PID: A16297840)

2024-02-20

In today's class we will explore and analyze data from a published RNA-seq experiment where airway smooth muscle cells were treated with dexamethasone, a synthetic glucocorticoid steroid with anti-inflammatory effects (Himes et al. 2014).

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

library(BiocManager)
library(DESeq2)
library(dplyr)
library(ggplot2)
```

Data Import

We have 2 input files: "count data" and "col data"

```
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
```

Q1. How many genes are in this dataset?

```
str(counts)
```

```
'data.frame':   38694 obs. of  8 variables:
 $ SRR1039508: num  723 0 467 347 96 ...
 $ SRR1039509: num  486 0 523 258 81 ...
 $ SRR1039512: num  904 0 616 364 73 1 6000 2640 692 531 ...
 $ SRR1039513: num  445 0 371 237 66 ...
 $ SRR1039516: num  1170 0 582 318 118 ...
 $ SRR1039517: num  1097 0 781 447 94 ...
 $ SRR1039520: num  806 0 417 330 102 ...
 $ SRR1039521: num  604 0 509 324 74 ...
```

38694 genes; there are 38694 observations in the counts df.

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

```
sum(metadata['dex'] == "control")
```

```
[1] 4
```

We have 4 control cell lines.

4. Toy differential gene expression

Analysis 4 treated, 4 control samples/experiments/columns.

Make sure the counts columns line up with the rows of the metadata.

```
colnames(counts) == metadata$id
```

```
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

To check that all elements of a vector are TRUE, we can use the `all()` function

```
all(colnames(counts) == metadata$id)
```

```
[1] TRUE
```

To start, I will calculate the `control.mean` values and `treated.mean` values and compare them.

- Identify and extract the `control` only columns
- Determine the mean value for each gene(i.e. row)
- Do the same for `treated`

```
control.inds <- metadata$dex == 'control'
control.counts <- counts[,control.inds]
control.mean <- apply(control.counts, 1, mean)
head(control.mean)
```

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

```
treated.inds <- metadata$dex == 'treated'
treated.counts <- counts[,treated.inds]
treated.mean <- apply(treated.counts, 1, mean)
head(treated.mean)
```

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

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

We can use `rowSums()`

```
control <- metadata %>% filter(dex=="control")
control.counts <- counts %>% select(control$id)
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
```

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)

```
treated <- metadata %>% filter(dex!="control")
treated.counts <- counts %>% select(treated$id)
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
```

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

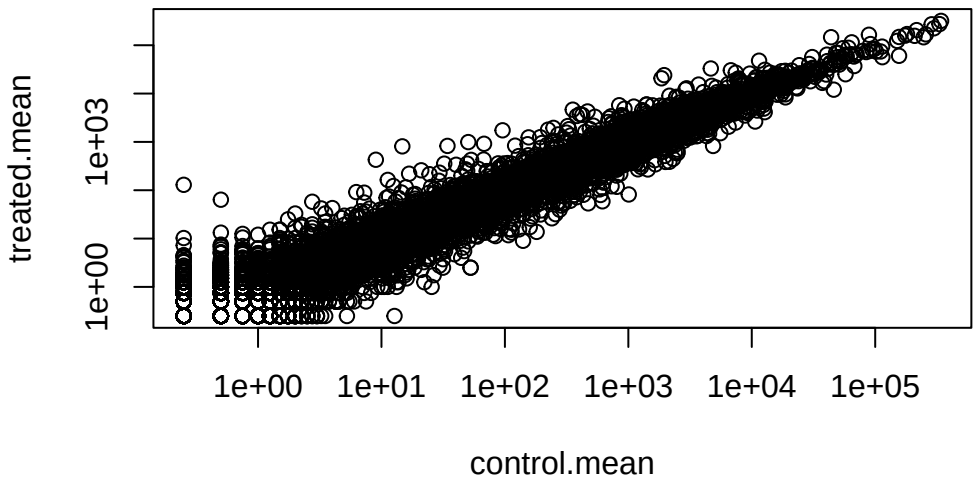
Have a quick view of this data:

Q5.a

```
plot(meancounts, log='xy')
```

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

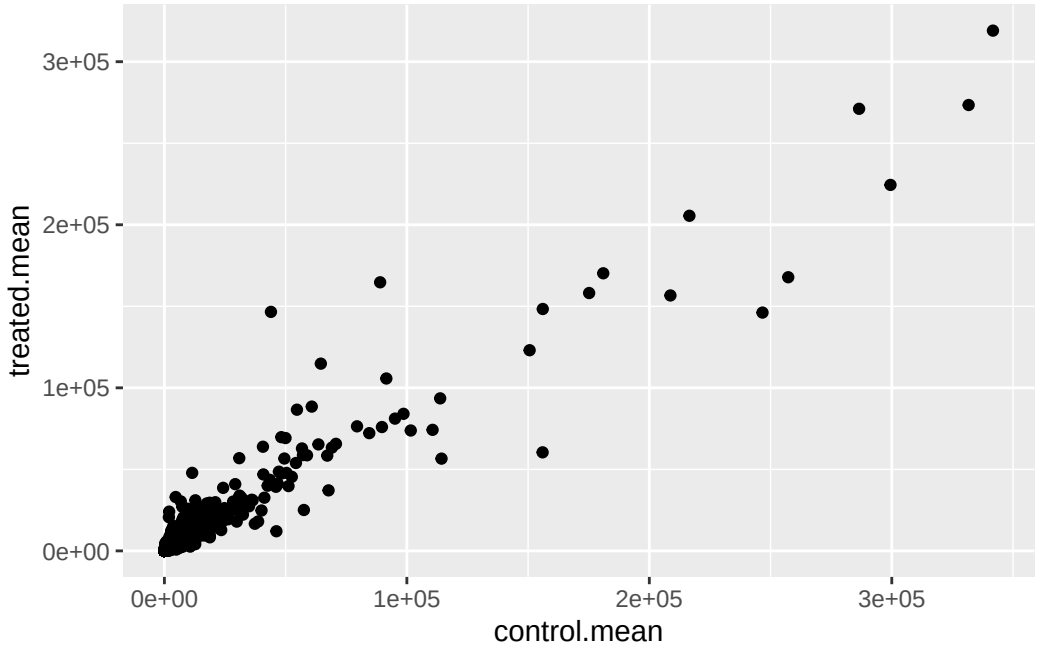
```
Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted
from logarithmic plot
```



Q5 (b).You could also use the ggplot2 package to make this figure producing the plot below. What geom_?() function would you use for this plot?

geom_point

```
ggplot(meancounts) + aes(x=control.mean, y=treated.mean) + geom_point()
```



I want to compare the treated and the control values here and we will use Fold change in log2 units to do this. log2(Treated/Control)

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

A common rule of thumb cutoff for calling a gene "differentially expressed" is a log2 fold-change value of either > +2 for upregulation or < -2 for downregulation

```
meancounts$log2fc <- log2fc
```

```
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
```

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

```
[1] 1846
```

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

```
[1] 2212
```

We first need to remove zero count genes as we can't say anything about these genes anyway and their division of log values are messing things up (divide by 0) or the -infinity log problem.

```
to.rm.ind <- rowSums(meancounts[,1:2]==0) > 0
mycounts <- meancounts[!to.rm.ind, ]
```

> Q. How many genes do we have left that we can say something about? (i.e. they don't have any 0 counts)

```
nrow(mycounts)
```

```
[1] 21817
```

```
up.ind <- mycounts[,3] > 2
down.ind <- mycounts[,3] < -2
sum(up.ind)
```

```
[1] 250
```

```
sum(down.ind)
```

```
[1] 367
```

> Q8. Using the up.ind vector above can you determine how many up regulated genes we have at the greater than 2 fc level?

250 genes

> Q9. Using the down.ind vector above can you determine how many down regulated genes we have at the greater than 2 fc level?

367 genes

Q10. Do you trust these results? Why or why not?

No, we need to see if the difference in the mean expression levels between the treated and control groups is significant.

DESeq analysis

Let's do this properly with the help of the DESeq2 package.

```
library(DESeq2)
```

We have to use a specific data object for working with DESeq.

```
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
```

Run our main analysis with 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)
```

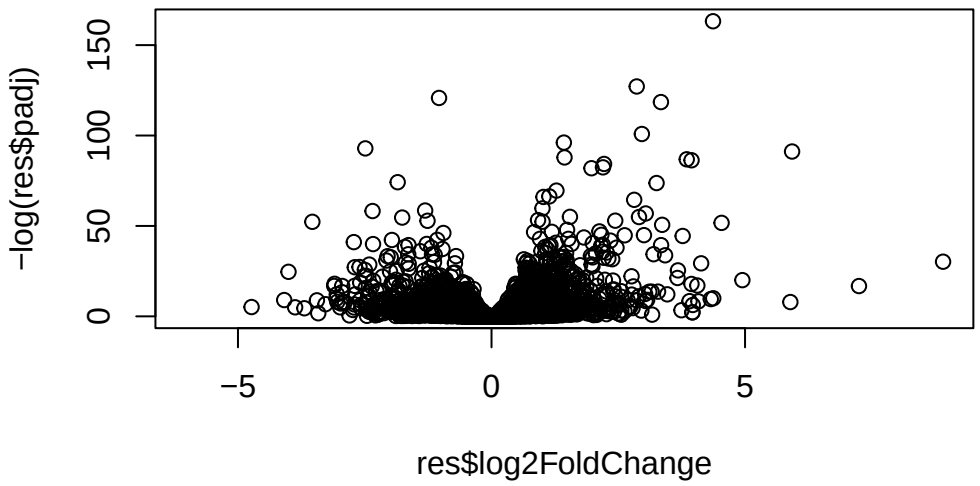
```
head(res)
```

```
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

A very common and useful summary results figure from this type of analysis is called a volcano plot - a plot of log2FC vs Adjusted P-Value. We use the `padj` for multiple testing.

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



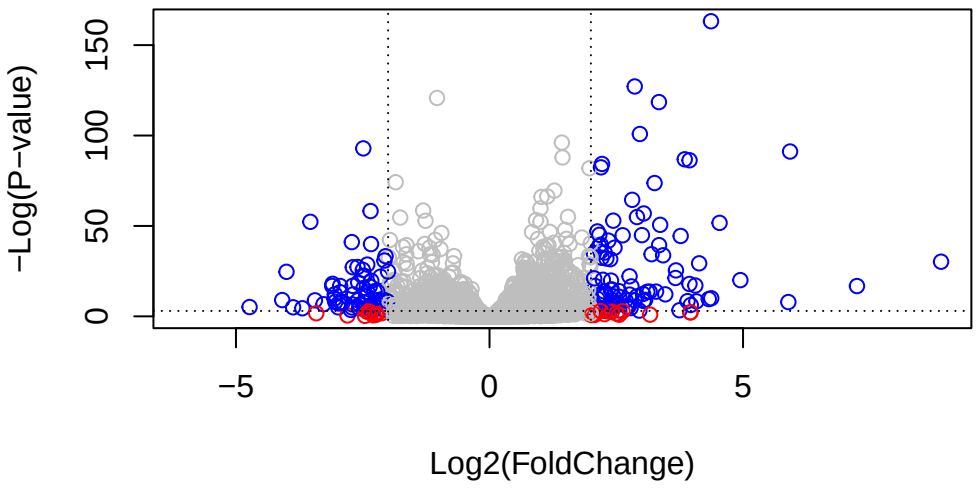
Smaller P value is more extreme log value

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

inds <- (res$padj < 0.05) & (abs(res$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="black", lty=3)
abline(h=-log(0.05), col="black", lty=3)
```



Add Annotation Data

We will use one of Bioconductor's main annotation packages to help with mapping between various ID schemes. Here we load the AnnotationDbi package and the annotation data package for humans org.Hs.eg.db.

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

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

```
 [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
 [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"
```

```
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
```

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

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

```
  res$genename <- mapIds(org.Hs.eg.db,
                    keys=row.names(res),      # Our genenames
```

```
                     keytype="ENSEMBL",        # The format of our genenames
                     column="GENENAME",          # 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 9 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      entrez              genename
                <numeric> <character> <character>           <character>
ENSG00000000003  0.163035      TSPAN6        7105          tetraspanin 6
ENSG00000000005        NA        TNMD       64102            tenomodulin
ENSG00000000419  0.176032        DPM1        8813 dolichyl-phosphate m..
ENSG00000000457  0.961694       SCYL3       57147 SCY1 like pseudokina..
ENSG00000000460  0.815849       FIRRM       55732 FIGNL1 interacting r..
ENSG00000000938        NA         FGR        2268 FGR proto-oncogene, ..
```

Pathway Analysis

Now that I have added the necessary annotation data, I can talk to different databases that use these IDs.

We will use the **gage** package to do geneset analysis (aka pathway analysis, geneset enrichment, overlap analysis)

```
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)
```

We will use KEGG first ()

```
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"
```

The main `gage()` function requires a named vector of fold changes, where the names of the values are the Entrez gene IDs.

```
  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
```

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

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

```
  # Look at the first three down (less) pathways
  head(keggres$less, 3)
```

```
                                     p.geomean stat.mean        p.val
hsa05332 Graft-versus-host disease 0.0004250461 -3.473346 0.0004250461
hsa04940 Type I diabetes mellitus  0.0017820293 -3.002352 0.0017820293
hsa05310 Asthma                    0.0020045888 -3.009050 0.0020045888
                                        q.val set.size         exp1
hsa05332 Graft-versus-host disease 0.09053483       40 0.0004250461
hsa04940 Type I diabetes mellitus  0.14232581       42 0.0017820293
hsa05310 Asthma                    0.14232581       29 0.0020045888
```

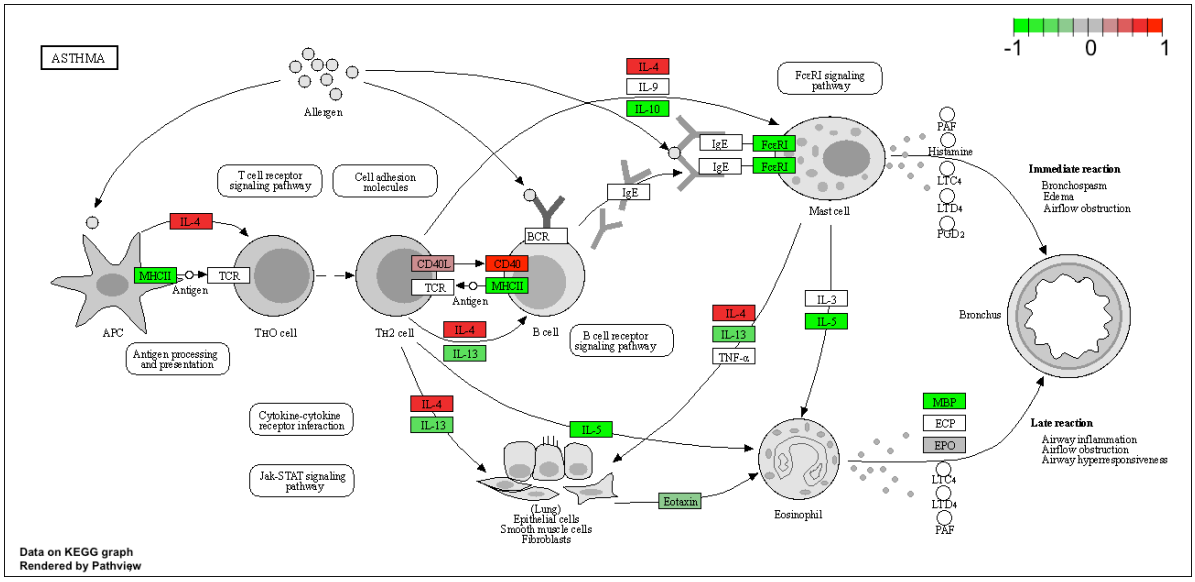
I can now use the return pathway IDs from KEGG as input to the `pathview` package to make pathway figures with our DEGs.

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

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

```
Info: Working in directory /Users/rahulnedunuri/Documents/ucsd courses/senior classes/winter
```

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


ASTHMA
-1
0
1
Allegen
IL-4
IL-9
IL-10
FcεRI signaling pathway
IgE
FcεRI
Mast cell
PAF
Histamine
LTC4
LTD4
PGD2
Immediate reaction
Bronchospasm
Edema
Airflow obstruction
T cell receptor signaling pathway
Cell adhesion molecules
BCR
MHCII
TCR
Antigen
APC
TH0 cell
TH2 cell
B cell
CD40L
CD40
IL-4
IL-13
B cell receptor signaling pathway
Antigen processing and presentation
IL-3
IL-5
IL-13
TNF-α
Bronchus
Cytokine-cytokine receptor interaction
Jak-STAT signaling pathway
(Lung)
Epithelial cells
Smooth muscle cells
Fibroblasts
Eotaxin
Eosinophil
MBP
ECP
EPO
Late reaction
Airway inflamnation
Airflow obstruction
Airway hyperresponsiveness
Data on KEGG graph
Rendered by Pathview