lab_12_RNASeq

Ryan Bench (PID: A69038034)

Table of contents

Background

Today we will analyze some RNASeq data from Himes et al. on the effects of a common steroid (dexmethasome, also called "dex") on airway smooth muscle cells (ASMs).

For this analysis we need two main inputs

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

Data Imports

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

Let's have a wee peak at our `counts`

```
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`

```
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?

```
nrow(counts)
```

```
[1] 38694
```

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

```
ncol(counts)
```

```
[1] 8
```

Q3. How many "control" experiments are there in the dataset?

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

```
[1] 4
```

Toy analysis example

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. 5. Compare these mean values for each gene.

Step 1.

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

Step 2.

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

Step 3-4

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

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

For ease of plotting and bookeeping, we can store these together in one data frame called `meancounts`

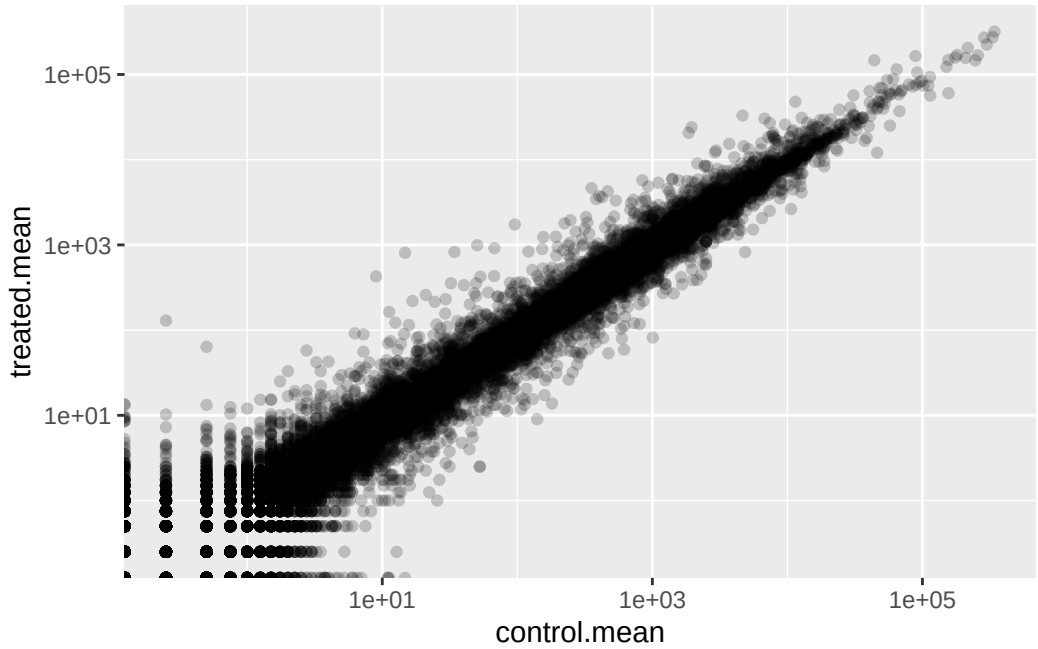
```
meancounts <- data.frame(control.mean, treated.mean)
head(meancounts)
```

```
                control.mean treated.mean
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
```

```
library(ggplot2)
ggplot(meancounts) +
  aes(x = control.mean, y = treated.mean) +
  geom_point(alpha = 0.2) +
  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.
```



This data is screaming to be log transformed

Treated/control is fold change

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

Log2 of 1 is 0

```
#treated/control
log2(10/10)
```

```
[1] 0
```

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

```
[1] 1
```

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

```
[1] -1
```

```
log2(40/10)
```

```
[1] 2
```

```
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
```

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

```
nonzero.inds <- rowSums(counts) !=0
mycounts <- meancounts[nonzero.inds, ]

#Alternate method find zero values
```

```
zero.inds <- which(meancounts[,1:2] == 0, arr.ind=TRUE)[,1]
mygenes <- meancounts[-zero.inds,]
```

Q3. How mnay genes are "up" regulated at the +2 log2FC threhsold?

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

```
[1] 314
```

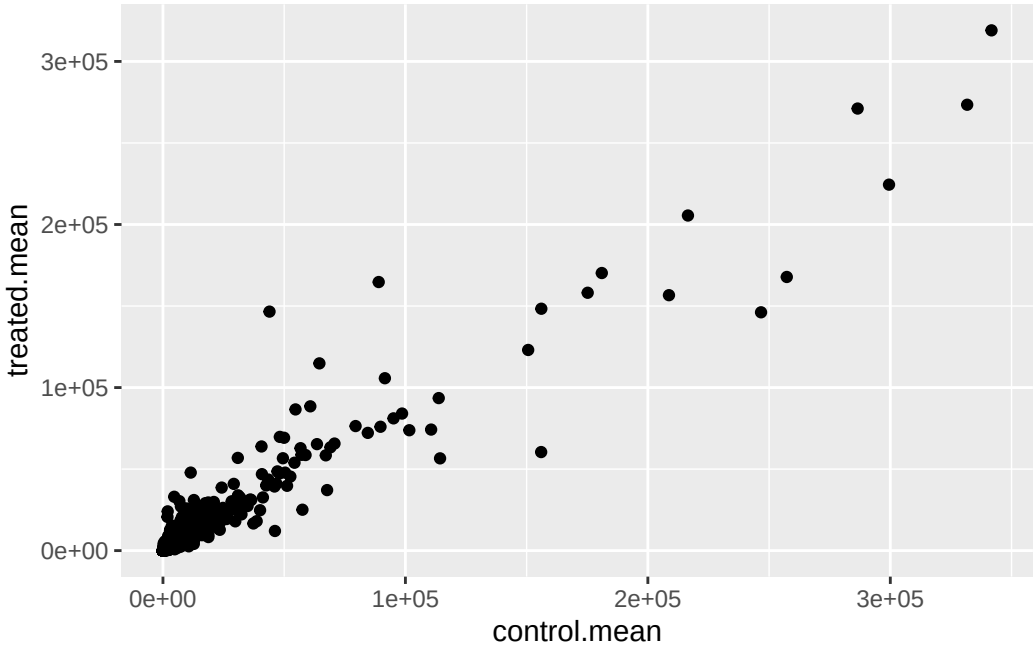
Q4. How many genes are "down" regulated (at the -2 log2FC threshold)?

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

```
[1] 485
```

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.

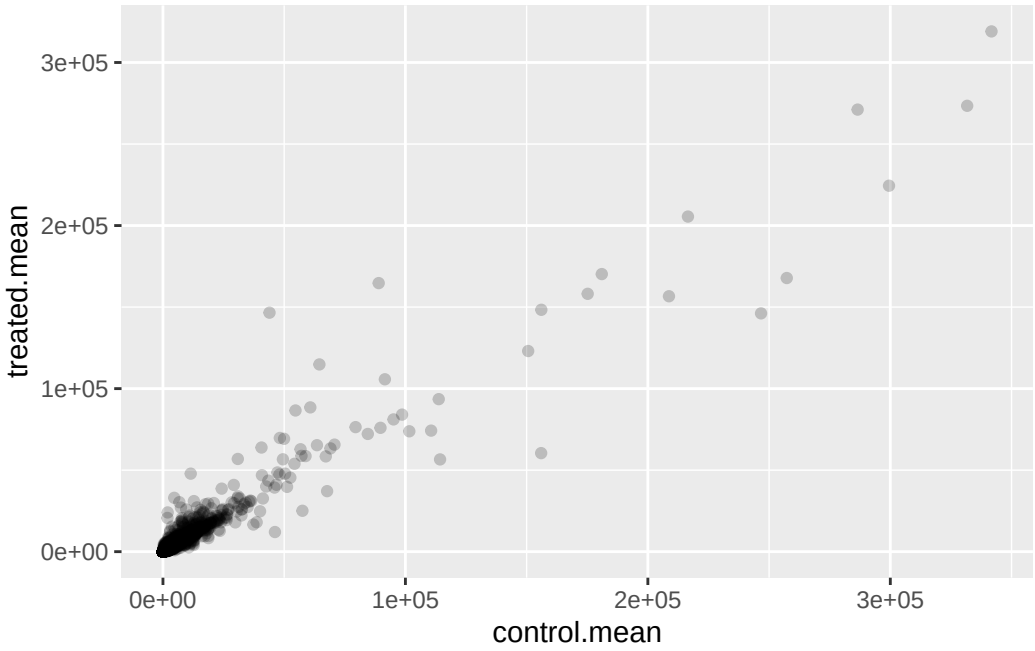
```
ggplot(meancounts) +
  aes(x = control.mean, y = treated.mean) +
  geom_point()
```



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?

I would adjust the alpha level (transparency) to 0.2.

```
library(ggplot2)
ggplot(meancounts) +
  aes(x = control.mean, y = treated.mean) +
  geom_point(alpha = 0.2)
```

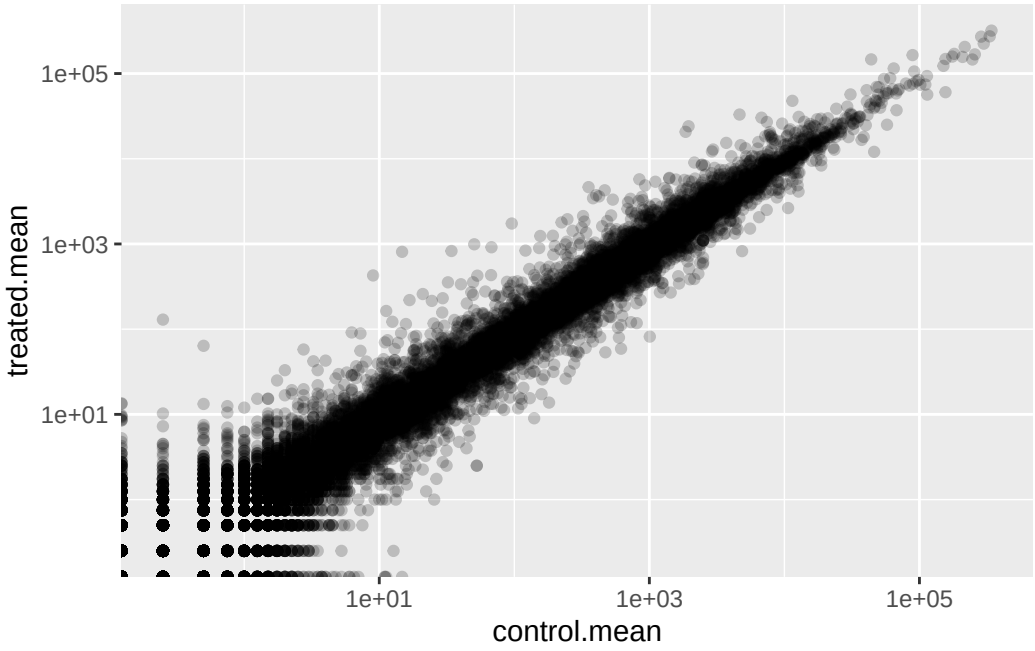


Q6. Try plotting both axes on a log scale. What is the argument to plot() that allows you to do this?

```
library(ggplot2)
ggplot(meancounts) +
  aes(x = control.mean, y = treated.mean) +
  geom_point(alpha = 0.2) +
  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.
```



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

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

```
[1] 314
```

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

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

```
[1] 485
```

DESeq Analysis

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

```
library(DESeq2)
```

SESeq wants three things for analysis, countData, colData, and design.

```
dds <- DESeqDataSetFromMatrix(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 in the SESeq package to run analysis is called `DESeq()`.

```
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()`

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

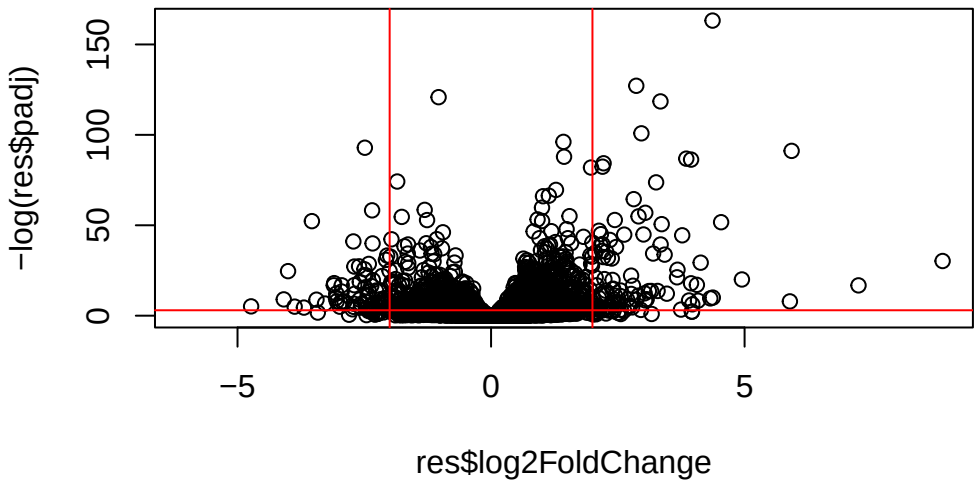
```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 6 columns
                  baseMean log2FoldChange     lfcSE      stat    pvalue
                 <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003 747.194195 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

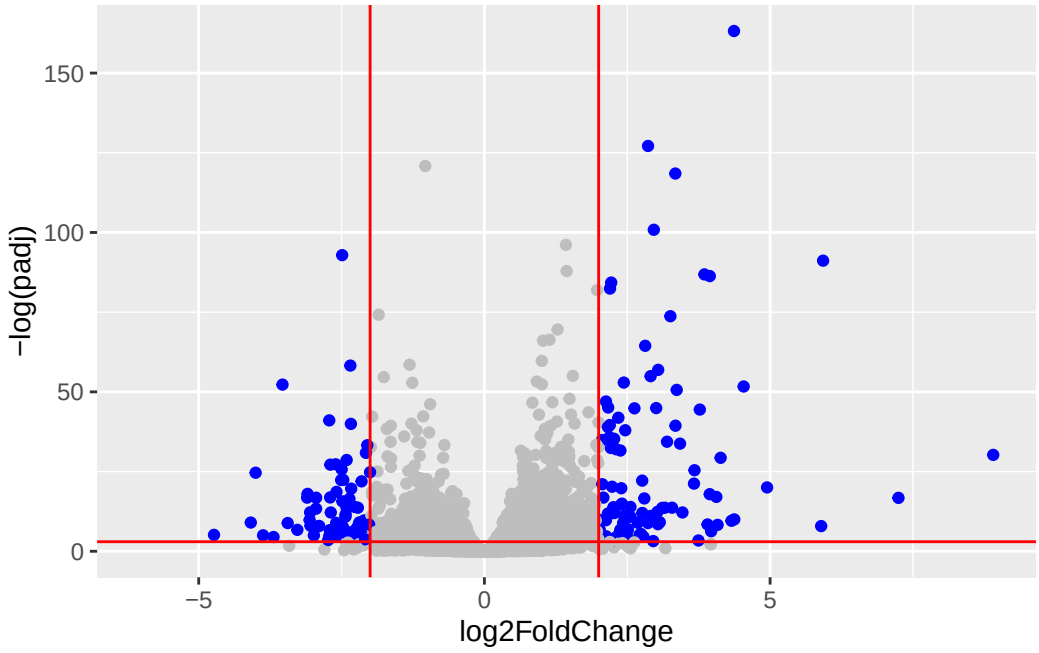


A nicer ggplot volcano plot

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

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

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



EnhancedVolcano would not work, I was troubleshooting for a very long time and it still would not work

```
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)
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"
```

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

```
[1] -0.35070296          NA  0.20610728  0.02452701 -0.14714263 -1.73228897
```

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

```
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
```

For some reason, I could not get this command to give me the right results either.

```
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/ryanbench/UCSD Classes/BGGN 213/R Stuff/Lab_12

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

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

```
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/ryanbench/UCSD Classes/BGGN 213/R Stuff/Lab_12

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

Save our results

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

##Add Annotation Data

We need to add gene symbols, gene names and other database IDs to make my results useful for further analysis

```
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
```

We have ENSEBLE database IDs in our `res` object

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

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

We can use the `mapIds()` function from bioconductor to help us.

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

Let's see what database id formats we can translate between

```
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),
                     keytype="ENSEMBL",
                     column="SYMBOL")
```

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

```
head(res$symbol)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
       "TSPAN6"          "TNMD"          "DPM1"         "SCYL3"         "FIRRM"
ENSG00000000938
          "FGR"
```

Add `GENENAME` then `ENTREZ`

```
res$genename <- mapIds(org.Hs.eg.db,
                       keys=row.names(res),
                       keytype="ENSEMBL",
                       column="SYMBOL")
```

```
'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 8 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      symbol    genename
                <numeric> <character> <character>
ENSG00000000003  0.163017      TSPAN6      TSPAN6
ENSG00000000005        NA        TNMD        TNMD
ENSG00000000419  0.175937        DPM1        DPM1
ENSG00000000457  0.961682       SCYL3       SCYL3
ENSG00000000460  0.815805       FIRRM       FIRRM
ENSG00000000938        NA         FGR         FGR
```

```
res$entrez <- mapIds(org.Hs.eg.db,
                     keys=row.names(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      -0.350703  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      symbol    genename      entrez
                <numeric> <character> <character> <character>
ENSG00000000003  0.163017      TSPAN6      TSPAN6        7105
ENSG00000000005        NA        TNMD        TNMD       64102
ENSG00000000419  0.175937        DPM1        DPM1        8813
ENSG00000000457  0.961682       SCYL3       SCYL3       57147
ENSG00000000460  0.815805       FIRRM       FIRRM       55732
ENSG00000000938        NA         FGR         FGR        2268
```

Save my annotated results

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

Pathway Analysis

We will use **gage** function from bioconductor.

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

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"
```

What **gage** wants as input is a simple named vector of importance i.e. a vector with labeled fold-changes

```
foldchanges <- res$log2FoldChange
names(foldchanges) <- res$entrez
head(foldchanges)
```

```
       7105       64102        8813       57147       55732        2268
-0.35070296          NA  0.20610728  0.02452701 -0.14714263 -1.73228897
```

Run gage analysis:

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

What is in the results:

```
attributes(keggres)
```

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

```
head(keggres$less, 5)
```

```
                                                      p.geomean stat.mean
hsa05332 Graft-versus-host disease                   0.0004250607 -3.473335
hsa04940 Type I diabetes mellitus                    0.0017820379 -3.002350
hsa05310 Asthma                                      0.0020046180 -3.009045
hsa04672 Intestinal immune network for IgA production 0.0060434609 -2.560546
hsa05330 Allograft rejection                         0.0073679547 -2.501416
                                                             p.val      q.val
hsa05332 Graft-versus-host disease                   0.0004250607 0.09053792
hsa04940 Type I diabetes mellitus                    0.0017820379 0.14232788
hsa05310 Asthma                                      0.0020046180 0.14232788
hsa04672 Intestinal immune network for IgA production 0.0060434609 0.31387487
hsa05330 Allograft rejection                         0.0073679547 0.31387487
                                                     set.size         exp1
hsa05332 Graft-versus-host disease                         40 0.0004250607
hsa04940 Type I diabetes mellitus                          42 0.0017820379
hsa05310 Asthma                                            29 0.0020046180
hsa04672 Intestinal immune network for IgA production      47 0.0060434609
hsa05330 Allograft rejection                               36 0.0073679547
```

Let's look at just one of these hsa05310

```
library(pathview)

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

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

Info: Working in directory /Users/ryanbench/UCSD Classes/BGGN 213/R Stuff/Lab_12

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

Insert figure for this pathway

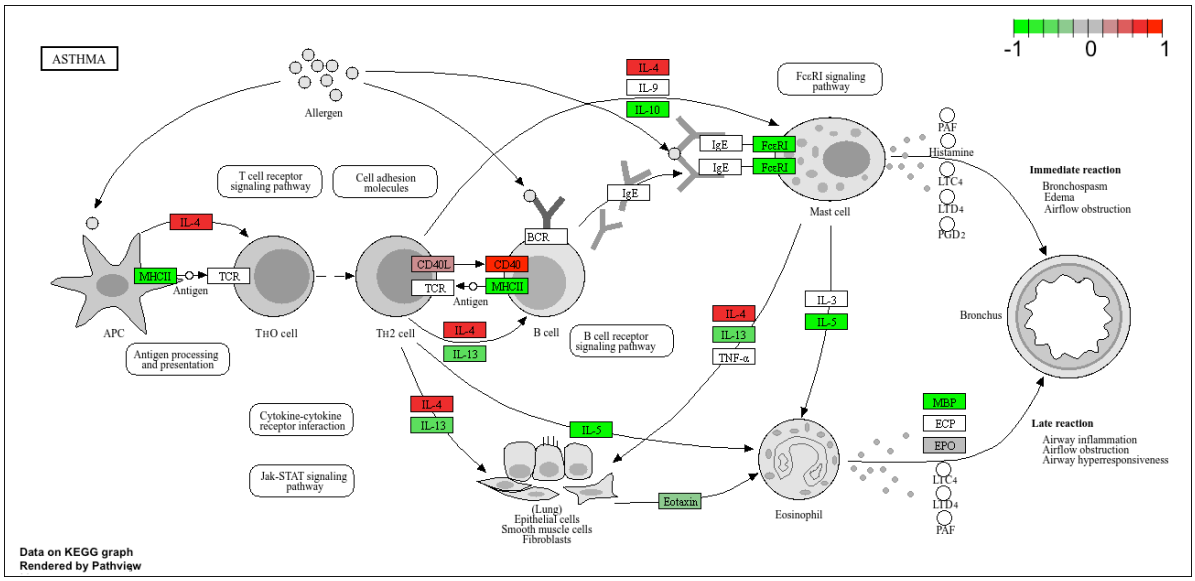


Figure 1: Asthma pathway from KEGG with my differentially expressed genes highlighted