Class 12: RNA Seq Analysis

Dan Vu (PID: A17380158)

Table of contents

Background

Today we will analyze some RNASeq data from Himes et al. on the effects of a common steroid on airway smooth muscle cells (ASM cells).

Our starting point is the "counts" data and "metadata" that contain the count values for each gene in their different experiments (i.e. cell lines with or without the drug)

Data import

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

Let's see what these datasets look like

```
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?

```
nrow(counts)
```

```
[1] 38694
```

Q2. How many different experiments are there? How many "control" cell lines are there?

```
nrow(metadata)
```

```
[1] 8
```

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

```
[1] 4
```

There are 4 experimental 'control' cell lines and 4 'control' cell lines.

Toy with differential gene expression

To start our analysis let's calculate the mean counts for all genes in the "control" experiments.

1. Extract all "control" columns from the `counts` object.
2. Calculate the mean for all rows(i.e. genes) of these "control" columns.

3-4. Do the same for "treated" 5. Compare these with `control.mean` values to see if theres any significant difference.

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

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

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

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

Yes, the above code could be more robust with the function `rowMeans`.

> 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)

Now let's do the same for the experimental cells.

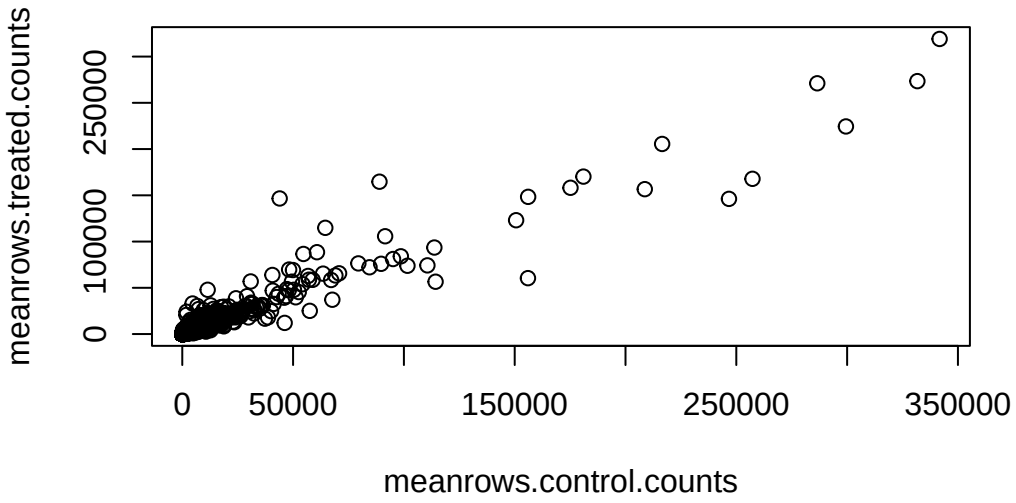
```
treated.inds <- metadata$dex == "treated"
treated.counts <- counts[,treated.inds]
```

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

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

```
                meanrows.control.counts meanrows.treated.counts
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
```

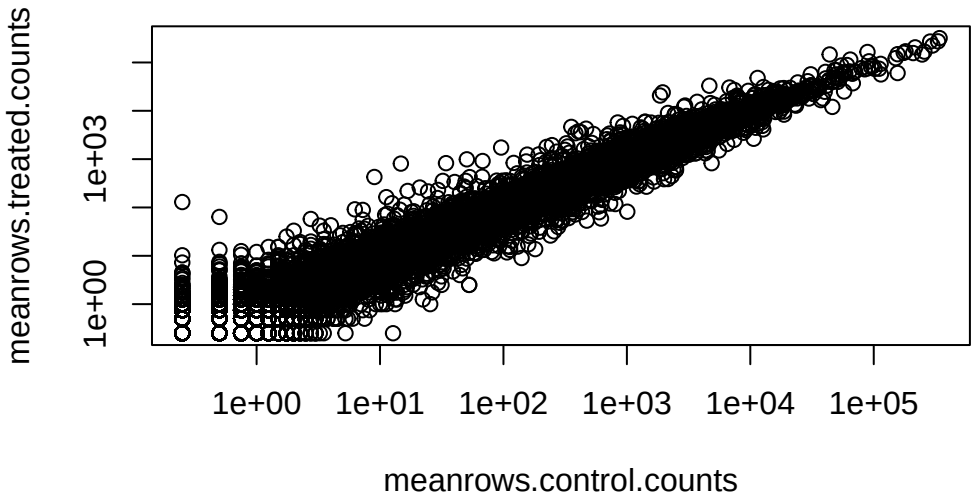
```
plot(meancounts)
```




```
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.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 use the `geom_point()` function.

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

The argument is `plot(meancounts, log="xy")`.

We often talk metrics like "log2 fold-change"

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

```
[1] 0
```

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

```
[1] -2
```

Let's calculate the log2 fold change for our treated over control mean counts.

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

```
head(meancounts)
```

```
                meanrows.control.counts meanrows.treated.counts      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
```

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

to.rm <- unique(zero.vals[,1])
mycounts <- meancounts[-to.rm,]
head(mycounts)
```

```
                meanrows.control.counts meanrows.treated.counts      log2fc
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
```

> Q7. What is the purpose of the arr.ind argument in the which() function call above? Why would we then take the first column of the output and need to call the unique() function?

The purpose of the `arr.ind` argument in the which() function is to report the positions where there are any "TRUE" values, allowing us to ignore any zero counts.

A common "rule of thumb" is a log2 fold change cutoff of +2 or -2 to call genes "Up regulated" or "Down regulated".

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

This is the number of up-regulated genes.

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

```
[1] 1910
```

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

This is the number of down-regulated genes.

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

```
[1] 2330
```

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

Although these fold changes are large, whether or not they are considered statistically significant between the two types of cell lines is still to be determined, so these results cannot be trusted.

DESeq2 Analysis

```
library(DESeq2)
```

For DESeq2 analysis we need three things. - count values (contData) - metadata telling us about the columns in `countData` (`colData`) - design of the experiment (i.e. what do you want to compare)

Our first function from DESeq2 will setup the input required for analysis by storing all these 3 things together.

```
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
```

The main function in DESeq2 that runs the 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
```

```
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
```

Adding Annotation Data

```
library("AnnotationDbi")
```

```
Attaching package: 'AnnotationDbi'

The following object is masked from 'package:dplyr':

    select
```

```
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      -0.350703  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
                <numeric> <character>
ENSG00000000003  0.163017      TSPAN6
ENSG00000000005        NA        TNMD
ENSG00000000419  0.175937        DPM1
ENSG00000000457  0.961682       SCYL3
ENSG00000000460  0.815805       FIRRM
ENSG00000000938        NA         FGR
```

Q11. Run the mapIds() function two more times to add the Entrez ID and UniProt accession and GENENAME as new columns called res$entrez, res$uniprot and res$genename.

```
res$entrez <- mapIds(org.Hs.eg.db,
                     keys=row.names(res),
                     column="ENTREZID",
                     keytype="ENSEMBL",
                     multiVals="first")
```

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

```
res$uniprot <- mapIds(org.Hs.eg.db,
                     keys=row.names(res),
                     column="UNIPROT",
                     keytype="ENSEMBL",
                     multiVals="first")
```

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

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

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

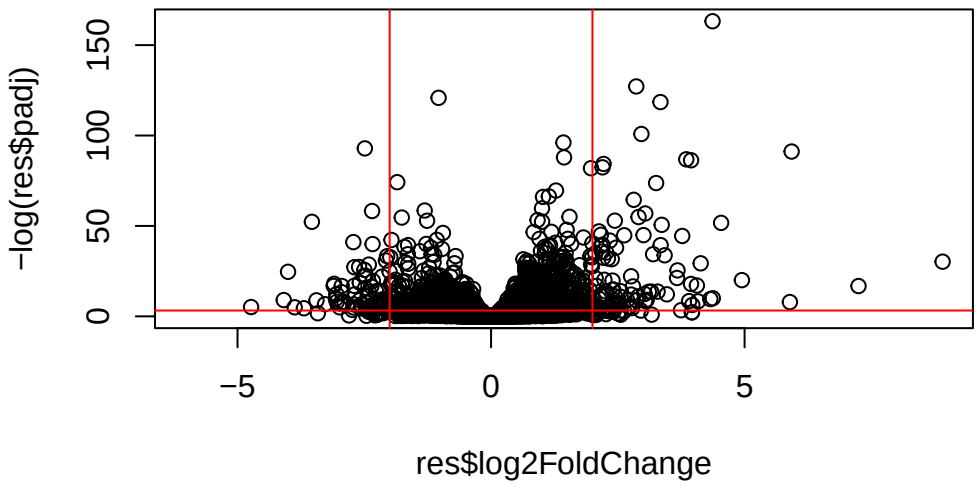
```
head(res)
```

```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 10 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      entrez     uniprot
                <numeric> <character> <character> <character>
ENSG00000000003  0.163017      TSPAN6        7105  A0A087WYV6
ENSG00000000005        NA        TNMD       64102      Q9H2S6
ENSG00000000419  0.175937        DPM1        8813      H0Y368
ENSG00000000457  0.961682       SCYL3       57147      X6RHX1
ENSG00000000460  0.815805       FIRRM       55732      A6NFP1
ENSG00000000938        NA         FGR        2268      B7Z6W7
                           genename
                        <character>
ENSG00000000003          tetraspanin 6
ENSG00000000005            tenomodulin
ENSG00000000419 dolichyl-phosphate m..
ENSG00000000457 SCY1 like pseudokina..
ENSG00000000460 FIGNL1 interacting r..
ENSG00000000938 FGR proto-oncogene, ..
```

Volcano Plot

This is a common summary result figure from these types of experiments and plot the log2 fold-change vs the p-value

```
plot(res$log2FoldChange, -log(res$padj))
abline(v=c(-2,2), col="red")
abline(h=-log(0.04), col="red")
```



Save our results

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

Add gene annotation

To help make sense of our results and communicate them to other folks we need to addd some more annotation to our main `res` object.

We will use two packages to first map IDs to different formats including the classic gene "symbol" gene name.

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

Let's see what is in the second package and what exactly it can do for us.

```
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"
```

We can translate or "map" IDs between any of these 26 databases using the `mapIDs()` function.

```
res$symbol <- mapIds(keys = row.names(res),
       keytype = "ENSEMBL",
       x = org.Hs.eg.db,
       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 10 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      entrez     uniprot
                <numeric> <character> <character> <character>
ENSG00000000003  0.163017      TSPAN6        7105  A0A087WYV6
ENSG00000000005        NA        TNMD       64102      Q9H2S6
ENSG00000000419  0.175937        DPM1        8813      H0Y368
ENSG00000000457  0.961682       SCYL3       57147      X6RHX1
ENSG00000000460  0.815805       FIRRM       55732      A6NFP1
ENSG00000000938        NA         FGR        2268      B7Z6W7
                          genename
                       <character>
```

```
ENSG00000000003          tetraspanin 6
ENSG00000000005            tenomodulin
ENSG00000000419 dolichyl-phosphate m..
ENSG00000000457 SCY1 like pseudokina..
ENSG00000000460 FIGNL1 interacting r..
ENSG00000000938 FGR proto-oncogene, ..
```

Add the mappings for "GENENAME" and "ENTREZID" and sore as `res$genename` and `res$entrez`.

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

```
'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 10 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      entrez     uniprot
                <numeric> <character> <character> <character>
ENSG00000000003  0.163017      TSPAN6        7105  A0A087WYV6
ENSG00000000005        NA        TNMD       64102      Q9H2S6
ENSG00000000419  0.175937        DPM1        8813      H0Y368
ENSG00000000457  0.961682       SCYL3       57147      X6RHX1
ENSG00000000460  0.815805       FIRRM       55732      A6NFP1
ENSG00000000938        NA         FGR        2268      B7Z6W7
                          genename
                       <character>
```

```
ENSG00000000003          tetraspanin 6
ENSG00000000005            tenomodulin
ENSG00000000419 dolichyl-phosphate m..
ENSG00000000457 SCY1 like pseudokina..
ENSG00000000460 FIGNL1 interacting r..
ENSG00000000938 FGR proto-oncogene, ..
```

```
res$entrezid <- mapIds(keys = row.names(res),
       keytype = "ENSEMBL",
       x = org.Hs.eg.db,
       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 11 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      entrez     uniprot
                <numeric> <character> <character> <character>
ENSG00000000003  0.163017      TSPAN6        7105  A0A087WYV6
ENSG00000000005        NA        TNMD       64102      Q9H2S6
ENSG00000000419  0.175937        DPM1        8813      H0Y368
ENSG00000000457  0.961682       SCYL3       57147      X6RHX1
ENSG00000000460  0.815805       FIRRM       55732      A6NFP1
ENSG00000000938        NA         FGR        2268      B7Z6W7
                           genename    entrezid
                        <character> <character>
ENSG00000000003          tetraspanin 6        7105
ENSG00000000005            tenomodulin       64102
ENSG00000000419 dolichyl-phosphate m..        8813
```

```
ENSG00000000457 SCY1 like pseudokina..       57147
ENSG00000000460 FIGNL1 interacting r..       55732
ENSG00000000938 FGR proto-oncogene, ..        2268
```

##Pathway Analysis

There are lots of bioconductor packages to do this type of analysis.

For now let's try one called **gage**.

```
library(gage)
```

```
library(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).
##############################################################################
```

To use **gage** I need two things.

- a names vector of fold-change values for our DEGs (our geneset of interest)
- a set of pathways or genesets to use for annotation

```
c("barry" = 5, "lisa" = 10)
```

```
barry  lisa
    5    10
```

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

```
     TSPAN6        TNMD        DPM1       SCYL3       FIRRM         FGR
-0.35070296          NA  0.20610728  0.02452701 -0.14714263 -1.73228897
```

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

In our results object we have:

```
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
```

Let's look at one of these pathways (hsa05310 Asthma) with our genes colored up so we can see the overlap.

```
pathview(pathway.id = "hsa05130", gene.data = foldchanges)
```

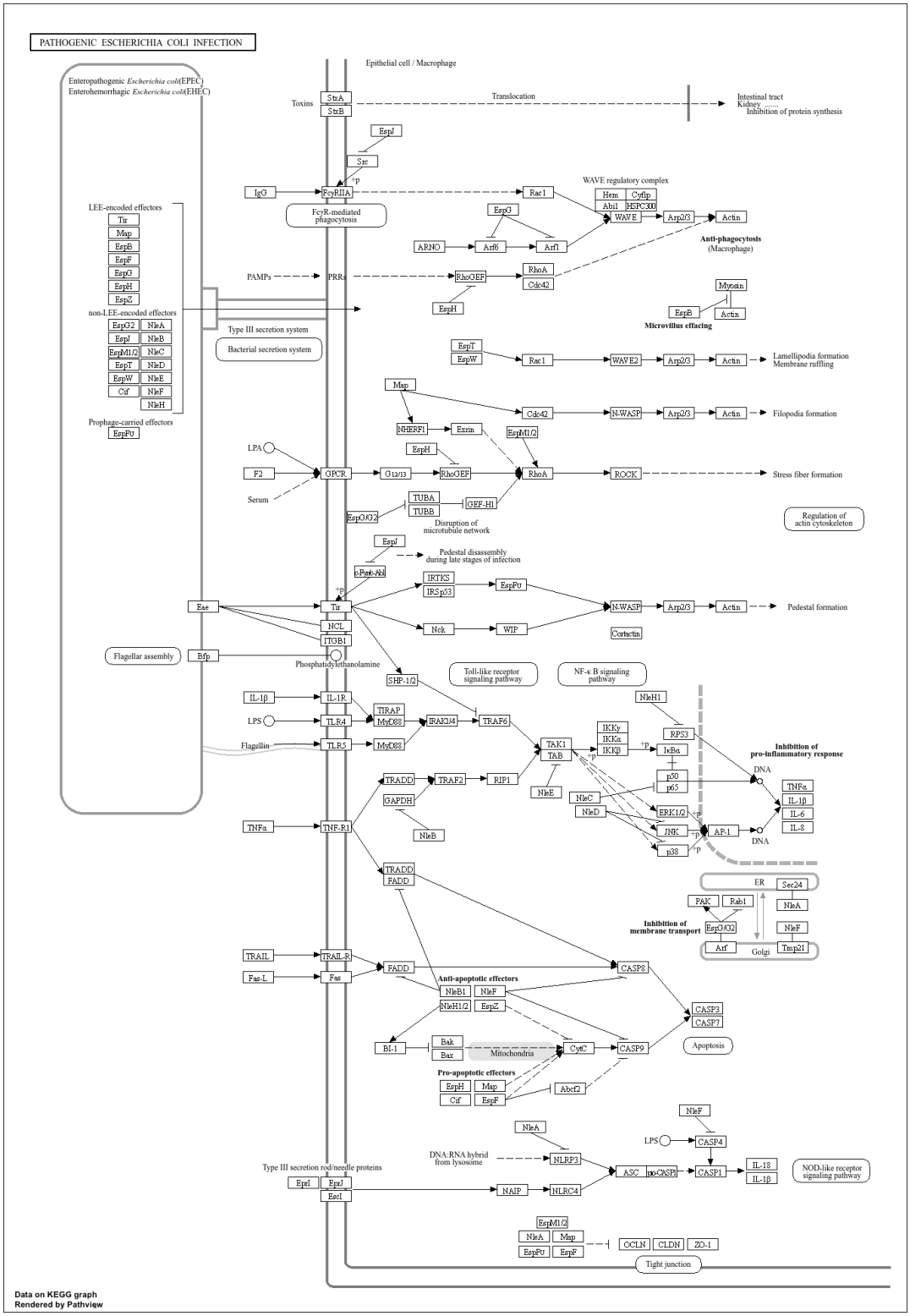
```
Warning: None of the genes or compounds mapped to the pathway!
Argument gene.idtype or cpd.idtype may be wrong.
```

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

```
Info: Working in directory /Users/danvu/Desktop/BIMM_143/Class_12
```

```
Info: Writing image file hsa05130.pathview.png
```

Add this pathway figure to our lab report.


PATHOGENIC ESCHERICHIA COLI INFECTION
Epithelial cell / Macrophage
Enteropathogenic Escherichia coli(EPEC)
Enterohemorrhagic Escherichia coli(EHEC)
Toxins
StxA
StxB
Translocation
Intestinal tract
Kidney .....
Inhibition of protein synthesis
EspJ
Src
IgG
FcγRIIA
FcγR-mediated phagocytosis
Rac1
WAVE regulatory complex
Hem
Cyfip
Abi1
HSPC300
WAVE
Arp2/3
Actin
EspG
ARNO
Arf6
Arf1
Anti-phagocytosis
(Macrophage)
LEE-encoded effectors
Tir
Map
EspB
EspF
EspG
EspH
EspZ
non-LEE-encoded effectors
EspG2
NleA
EspJ
NleB
EspM1/2
NleC
EspT
NleD
EspW
NleE
Cif
NleF
NleH
Prophage-carried effectors
EspFU
PAMPs
PRRs
RhoGEF
RhoA
Cdc42
EspH
Myosin
EspB
Actin
Microvillus effacing
Type III secretion system
Bacterial secretion system
EspT
EspW
Rac1
WAVE2
Arp2/3
Actin
Lamellipodia formation
Membrane ruffling
Map
Cdc42
N-WASP
Arp2/3
Actin
Filopodia formation
NHERF1
Ezrin
EspM1/2
LPA
F2
Serum
GPCR
G12/13
RhoGEF
RhoA
ROCK
Stress fiber formation
TUBA
TUBB
GEF-H1
EspG/O2
Disruption of microtubule network
Regulation of actin cytoskeleton
EspJ
Pedestal disassembly during late stages of infection
c-Fyn/c-Abl
IRTKS
IRSp53
EspFU
Eae
Tir
N-WASP
Arp2/3
Actin
Pedestal formation
NCL
ITGB1
Nck
WIP
Cortactin
Flagellar assembly
Bfp
Phosphatidylethanolamine
SHP-1/2
Toll-like receptor signaling pathway
NF-κ B signaling pathway
NleH1
IL-1β
IL-1R
LPS
TLR4
TIRAP
MyD88
IRAK1/4
TRAF6
Flagellin
TLR5
MyD88
TAK1
TAB
IKKγ
IKKα
IKKβ
IκBα
RPS3
Inhibition of pro-inflammatory response
DNA
p50
p65
TNFα
IL-1β
IL-6
IL-8
TRADD
TRAF2
RIP1
GAPDH
NleE
NleC
NleD
ERK1/2
JNK
p38
AP-1
DNA
TNFα
TNF-R1
NleB
TRADD
FADD
ER
Sec24
PAK
Rab1
NleA
Inhibition of membrane transport
EspG/O2
NleF
Arf
Golgi
Tmp21
TRAIL
Fas-L
TRAIL-R
Fas
FADD
CASP8
Anti-apoptotic effectors
NleB1
NleF
NleH1/2
EspZ
CASP3
CASP7
Bl-1
Bak
Bax
Mitochondria
CytC
CASP9
Apoptosis
Pro-apoptotic effectors
EspH
Map
Cif
EspF
Abcf2
NleF
NleA
LPS
CASP4
DNA:RNA hybrid from lysosome
NLRP3
ASC
pro-CASP1
CASP1
IL-18
IL-1β
NOD-like receptor signaling pathway
Type III secretion rod/needle proteins
EprI
EprJ
EscI
NAIP
NLRC4
EspM1/2
NleA
Map
EspFU
EspF
OCLN
CLDN
ZO-1
Tight junction
Data on KEGG graph
Rendered by Pathview