Lab 12

Here we will use the DESeq2 package for RNASeq analysis. The data for todays class comes from a study of airway smooth muscle cells treated with dexamethasone, a synthetic glucocorticoid steroid with anti-inflammatory effects (Himes et al. 2014)

Import the data

We need two things for analysis: - countData (counts for every transcript/ gene in each experiment) - colData (metadata that describes the experimental setup)

```
countData <- read.csv("airway_scaledcounts.csv", row.names = 1)
head(countData)
```

```
                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
```

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

```
               dex celltype     geo_id
SRR1039508 control   N61311 GSM1275862
SRR1039509 treated   N61311 GSM1275863
SRR1039512 control  N052611 GSM1275866
SRR1039513 treated  N052611 GSM1275867
SRR1039516 control  N080611 GSM1275870
SRR1039517 treated  N080611 GSM1275871
SRR1039520 control  N061011 GSM1275874
SRR1039521 treated  N061011 GSM1275875
```

Q1. How many genes are there in this dataset?

```
nrow(countData)
```

```
[1] 38694
```

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

```
#dex column in meta data tells us the control?
sum(metadata$dex == "control")
```

```
[1] 4
```

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

```
control treated
      4       4
```

- Step 1. Calculate the mean of the control samples (i.e. columns in countData) Calculate the mean of the treated samples

(a) We need to find which columns are "control" samples

- look in the metadata (aka. colData), $dex column

```
control.inds <- metadata$dex == "control"
#output: TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE

# return just the control columns
```

```
head(countData[ , control.inds])
```

```
                SRR1039508 SRR1039512 SRR1039516 SRR1039520
ENSG00000000003        723        904       1170        806
ENSG00000000005          0          0          0          0
ENSG00000000419        467        616        582        417
ENSG00000000457        347        364        318        330
ENSG00000000460         96         73        118        102
ENSG00000000938          0          1          2          0
```

(b) Extract all the control columns from `contData` and call it `control.counts`

```
control.counts <- countData[ , control.inds]
```

(c) Calculate the mean value across the rows of `control.counts` i.e. calculate the mean count values for each gene in the control samples.

```
control.means <- rowMeans(control.counts)
head(control.means)
```

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

Treated columns

```
treated.inds <- metadata$dex == "treated"
treated.counts <- countData[ , treated.inds]
treated.means <- rowMeans(treated.counts)
head(treated.means)
```

```
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?

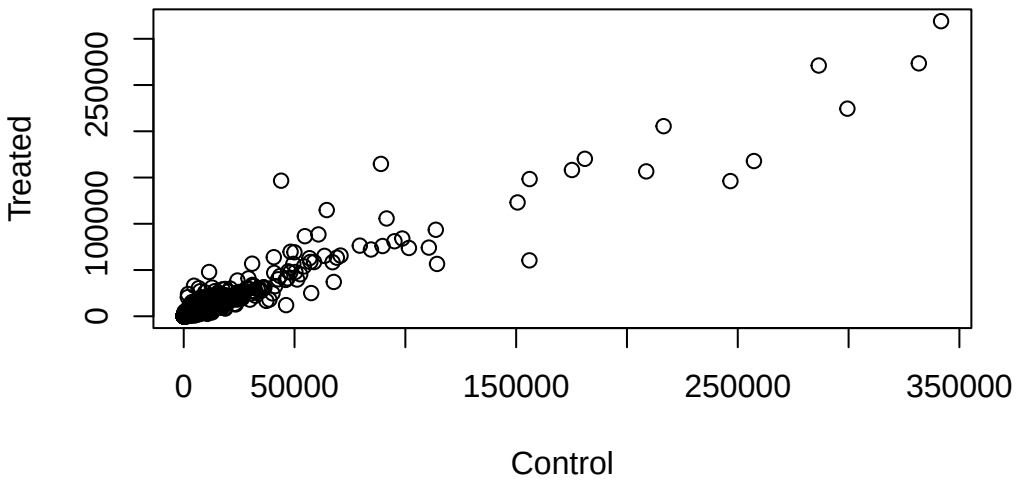
Q4. Follow the same procedure for the treated samples (i.e. calculate the mean per gene across drug treated samples and assign to a labeled vector called treated.mean)

combining mean data

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

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(meancounts[,1],meancounts[,2], xlab="Control", ylab="Treated")
```



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

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

We use log transforms for skewed data such as this and because we really are most about relative changes in magnitude.

We most often use log2 as our transform as the math is easier to interpret than log10 or others. If we have no change

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

```
[1] 0
```

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

```
[1] -1
```

```
#if I have half the maount I will have a log2 fold-chaneg of -1
log2(20/10)
```

```
[1] 1
```

```
#if I have double the amount i.e. 20 compared to 10 for examples I will have a log2 fold-c
```

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

```
                control.means treated.means      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
```

```
#if log2 fold-change of -2 or lower, its down regulated and if higher than 2 its up regula
```

Q8. How many genes are up-regulated at the common threshold of 2+ log2FC values?

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

```
[1] 1910
```

Q9. How many genes are down-regulated at the common threshold of 2+ log2FC values?

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

```
[1] 2330
```

Wait a minute...What about the stats! Are these changes significant? - To do this properly we will turn to DESeq2 package

DESeq Analysis

(Q10. We do not trust these results yet! Lets find the significance)

```
library(DESeq2)
# takes away long error messages/ the loading text when u access a package
```

To use DESeq we need our input countData and coldata in a specific format that DESeq wants:

```
dds <- DESeqDataSetFromMatrix(countData = countData,
                        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
```

To run the analysis I can now use the main DESeq2 function called `DESeq()` with `dds` as input

```
dds <- DESeq(dds)
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

To get results out of `dds` we can use `results()` function from package

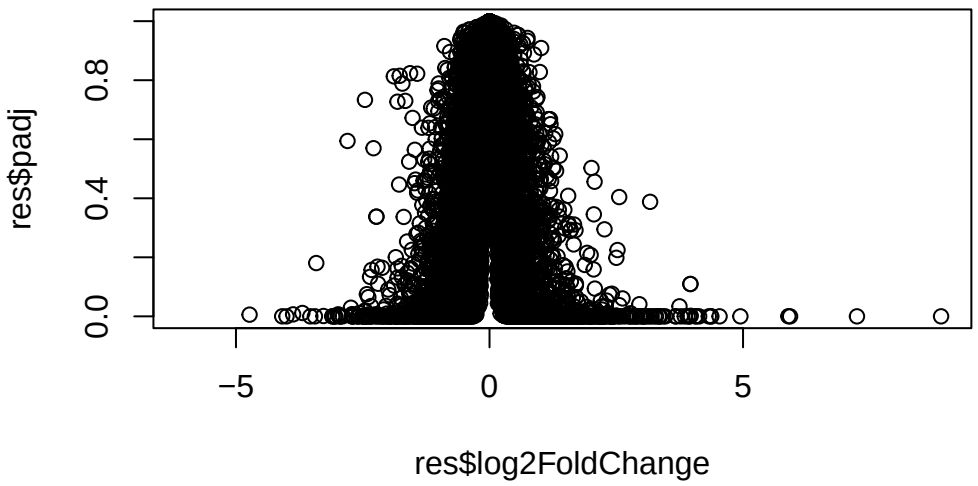
```
res <- results(dds)
head(res)
```

```
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
```

***padj**: adjustment of p-values for doing multiple tests*

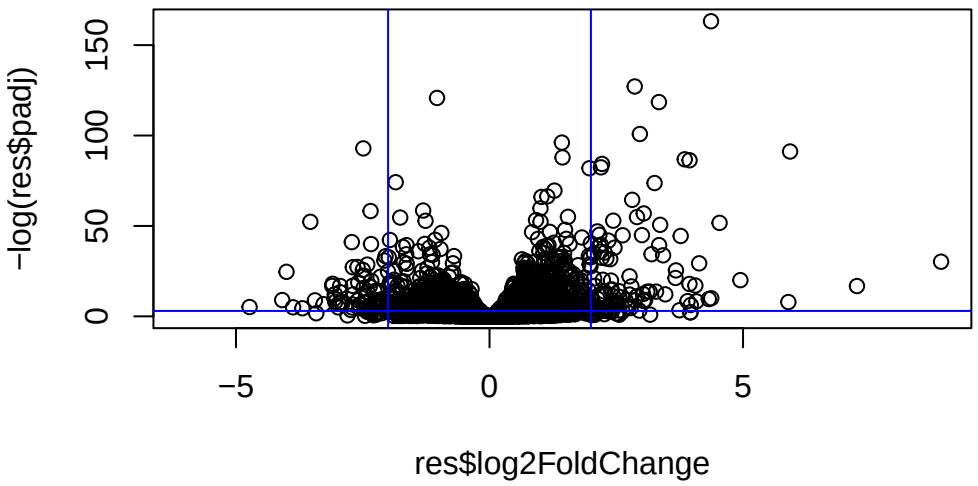
Let's make a final (for today) plot of log2 fold-change vs the adjusted P-value. ## Volcano Plot

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




```
# 0 means no change... change happens as we move away from 0. we care about the low p-valu
```

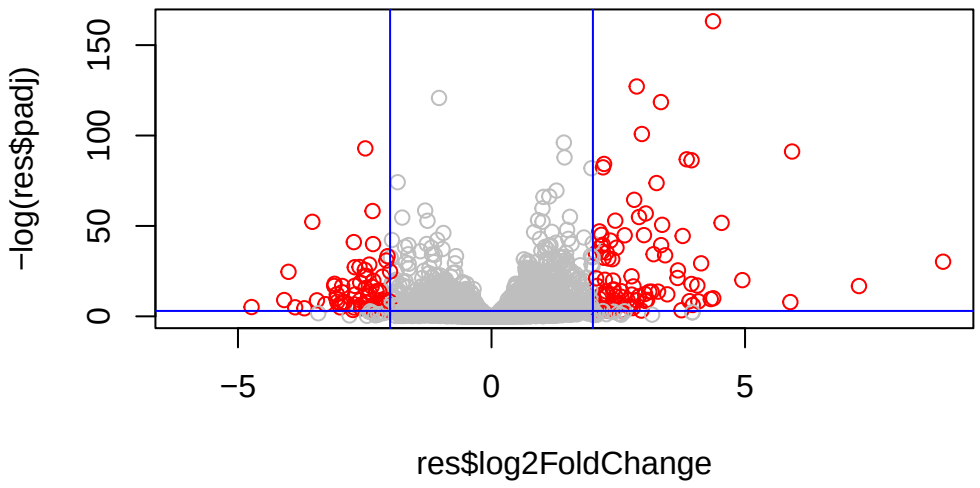
```
plot(res$log2FoldChange, -log(res$padj))
abline(v=c(+2,-2), col="blue")
abline(h=-log(0.05), col="blue")
```



Finally we can make our color vector to use in the plot to better highlight the genes we care about.

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

plot(res$log2FoldChange, -log(res$padj), col=mycols)
abline(v=c(+2,-2), col="blue")
abline(h=-log(0.05), col="blue")
```



Adding Annotation Data

We can use the AnnotationDbi package

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

We can translate (a.k.a. "map") between all these database id formats:

```
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", #format of genenames
```

```
                      column="SYMBOL", # ew format we want to add
                      multiVals="first")
```

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

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

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

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

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

write your own csv!!

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

Pathway Analysis

We can use the KEGG database of biological pathways to get some more insigth into out expressed genes and the kinds of biology they are involved in

```
library(pathview)
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"
```

```
foldchanges = res$log2FoldChange
names(foldchanges) = res$symbol
```

Example:

```
x <- 1:3
x
```

```
[1] 1 2 3
```

```
names(x) <- c("chandra", "lisa", "xinqui")
x
```

```
chandra    lisa  xinqui
      1       2       3
```

```
head(foldchanges)
```

```
     TSPAN6        TNMD        DPM1       SCYL3    C1orf112         FGR
-0.35070302          NA  0.20610777  0.02452695 -0.14714205 -1.73228897
```

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

Look at the top 3 "LESS"

```
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 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
```

```
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/anaghapashilkar/Desktop/school/BIMM 143/class 15/class 15
```

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

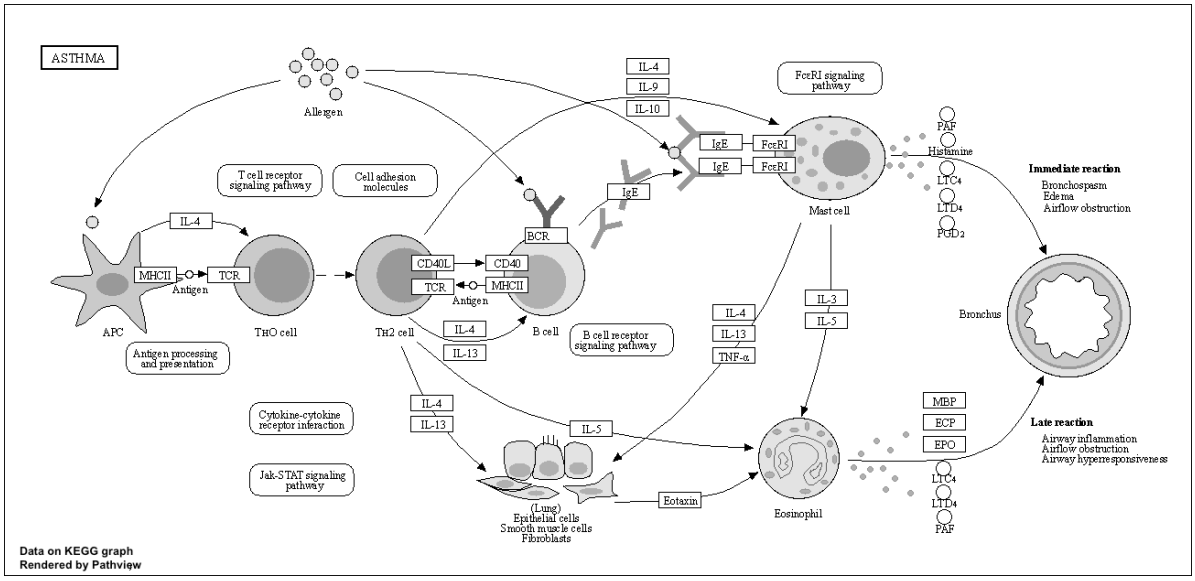


Figure 1: Asthma pathway from KEGG with our genes shown