Class 12: Transcriptomics and the analysis of RNA-Seq data

Libby Gilmore A69047570

Table of contents

Background

Today we will analyze some RNASeq data from Himes at al. on the effects of the common steroid (dexamethasone, 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 here must match the columns in `countData`

Bioconductor setup

```
Loading required package: S4Vectors
```

```
Loading required package: stats4
```

```
Loading required package: BiocGenerics

Loading required package: generics


Attaching package: 'generics'

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

    as.difftime, as.factor, as.ordered, intersect, is.element, setdiff,
    setequal, union


Attaching package: 'BiocGenerics'

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

    IQR, mad, sd, var, xtabs

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

    anyDuplicated, aperm, append, as.data.frame, basename, cbind,
    colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
    get, grep, grepl, is.unsorted, lapply, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, saveRDS, table, tapply, unique,
    unsplit, which.max, which.min


Attaching package: 'S4Vectors'

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

    findMatches

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

    expand.grid, I, unname

Loading required package: IRanges
```

```
Loading required package: GenomicRanges

Loading required package: Seqinfo

Loading required package: SummarizedExperiment

Loading required package: MatrixGenerics

Loading required package: matrixStats


Attaching package: 'MatrixGenerics'

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

    colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
    colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
    colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
    colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
    colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
    colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
    colWeightedMeans, colWeightedMedians, colWeightedSds,
    colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
    rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
    rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
    rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
    rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
    rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
    rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
    rowWeightedSds, rowWeightedVars

Loading required package: Biobase

Welcome to Bioconductor

    Vignettes contain introductory material; view with
    'browseVignettes()'. To cite Bioconductor, see
    'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
Attaching package: 'Biobase'

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

    rowMedians

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

    anyMissing, rowMedians


Attaching package: 'dplyr'

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

    combine

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

    count

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

    intersect, setdiff, union

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

    intersect

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

    collapse, desc, intersect, setdiff, slice, union

The following objects are masked from 'package:S4Vectors':

    first, intersect, rename, setdiff, setequal, union
```

```
The following objects are masked from 'package:BiocGenerics':

    combine, intersect, setdiff, setequal, union

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

    explain

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

    filter, lag

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

    intersect, setdiff, setequal, union
```

Data Import

```
# Complete the missing code
counts <- read.csv("airway_scaledcounts.csv", row.names=1)
metadata <-  read_csv("airway_metadata.csv")
```

```
Rows: 8 Columns: 4
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (4): id, dex, celltype, geo_id

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
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)
```

```
# A tibble: 6 x 4
  id         dex     celltype geo_id
  <chr>      <chr>   <chr>    <chr>
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`)?

```
ncol(counts)
```

```
[1] 8
```

Q3. How many 'control' cell lines do we have?

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

```
[1] 4
```

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

```
control treated
      4       4
```

Toy differential gene expression

Q4 Extract the "control" columns from `counts`. Calculate the mean value for each gene in these "control" columns

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

2. Calculate the mean value for each gene in these "control" columns

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

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

Q 5. Do the same with the "treated" columns

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

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

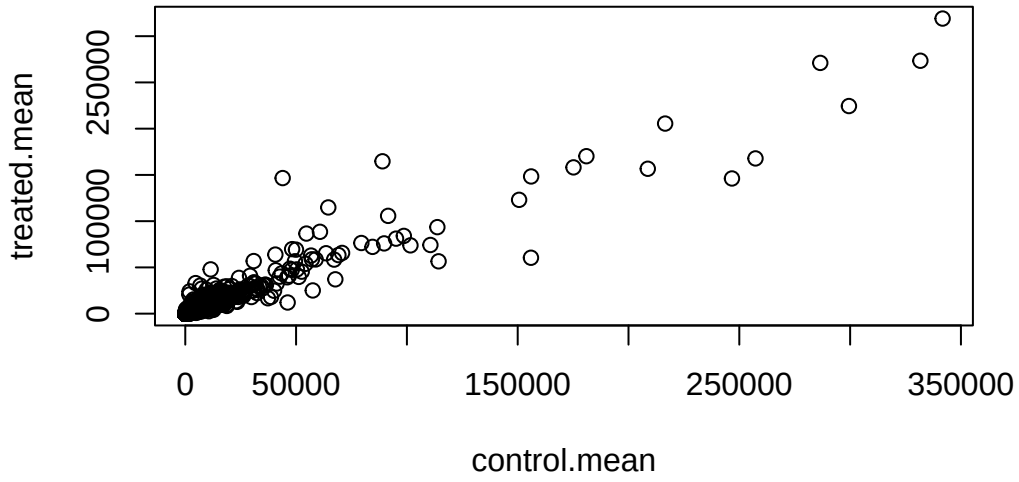
For ease of book-keeing we can store these together in one dataframe `meanCounts`

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.

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

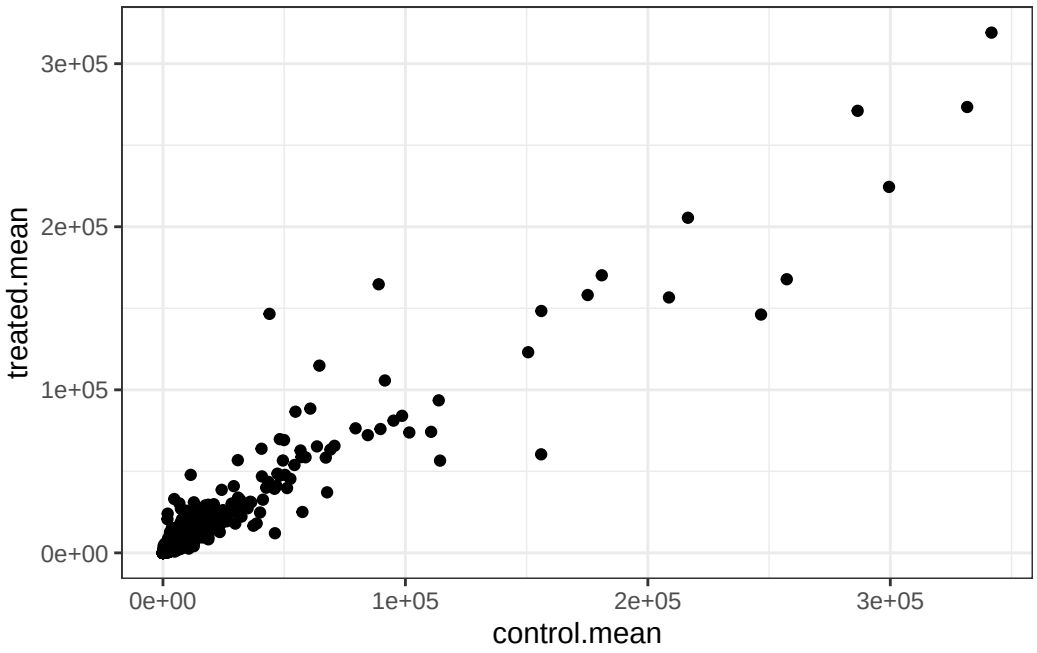
```
                 control.mean treated.mean
ENSG00000000003       900.75       658.00
ENSG00000000005         0.00         0.00
ENSG00000000419       520.50       546.00
ENSG00000000457       339.75       316.50
ENSG00000000460        97.25        78.75
ENSG00000000938         0.75         0.00
```

```
plot(meanCounts)
```



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?

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



Log transform the data

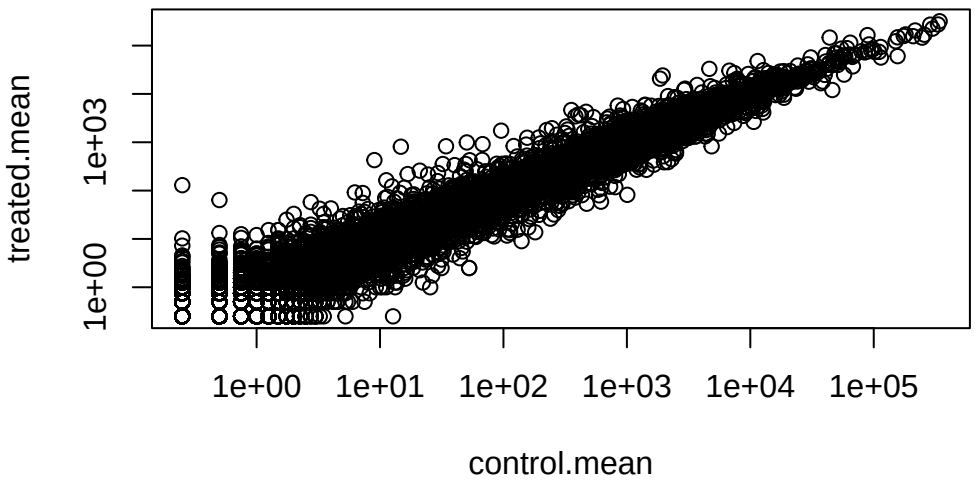
> Q6. Try plotting both axes on a log scale. What is the argument to plot() that allows you to do this?

`logxy` parameter

```
plot(meanCounts, log="xy")
```

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

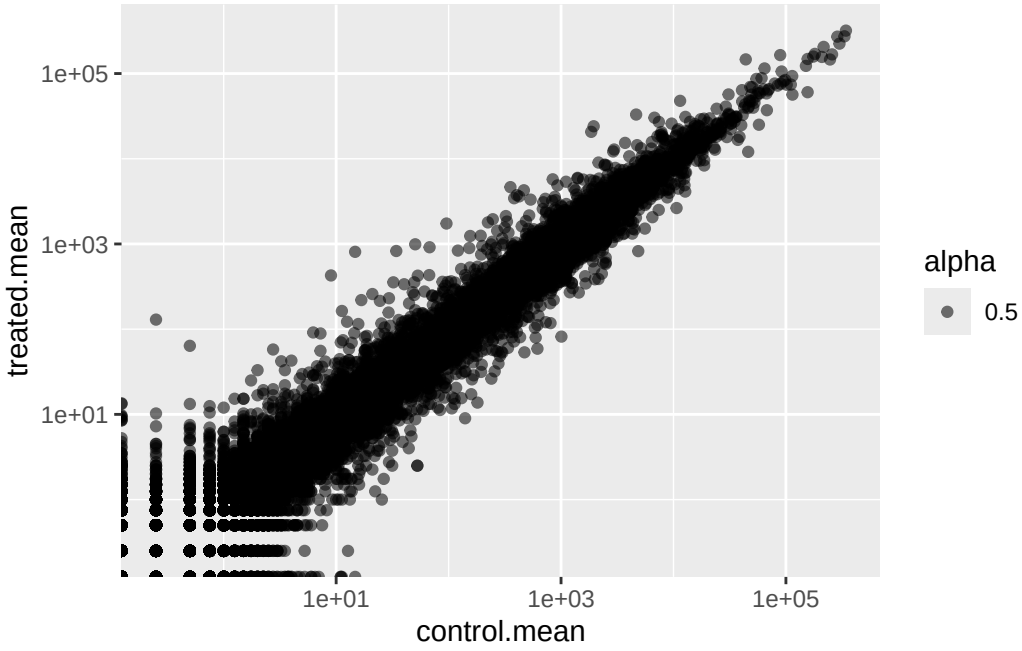
```
Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted
from logarithmic plot
```




```
ggplot(meanCounts, aes(x =control.mean, y=treated.mean, alpha = .5))+
  geom_point() +
  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.
```



We use log2 "fold change" as a way to compare

```
# treated/control
log2(10/10) # if the drug had no effect (1/1) than it is zero
```

```
[1] 0
```

```
log2(20/10) # if you have twice as much when the drug is around
```

```
[1] 1
```

```
log2(10/20) # if you have half as much transcript when drug is around
```

```
[1] -1
```

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

```
[1] 2
```

```
meanCounts <- meanCounts |>
  mutate(log2FC = log2(treated.mean/control.mean))

meanCounts$log2FC <- log2(meanCounts$treated.mean/meanCounts$control.mean)
```

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 lower.

Filter out zero count genes

> 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?

arr.ind tells you the row and column position

```
# y <- data.frame(a = c(1,5,0,5), b=c(1,0,5,5))
# which(y==0, arr.ind=T) # tells you row and column that is 0

zero.inds <- which(meanCounts[,1:2] == 0, arr.ind=T)[,1]
mygenes <- meanCounts[-zero.inds,]
sum(mygenes$log2FC >= 2)
```

```
[1] 314
```

```
sum(mygenes$log2FC <= -2)
```

```
[1] 485
```

> Q8 How many genes are "up" regulated at the +2 log2FC threshold.

```
meanCounts_noNA <- meanCounts |>
  filter(control.mean != 0.0 & treated.mean != 0.0)

sum(meanCounts_noNA$log2FC >= 2)
```

```
[1] 314
```

```
# alternative lab method:
# zero.vals <- which(meancounts[,1:2]==0, arr.ind=TRUE)
#
# to.rm <- unique(zero.vals[,1])
# mycounts <- meancounts[-to.rm,]
# head(mycounts)
```

Q9 How many genes are "down" regulated (at the -2 log2FC threshold)

```
sum(meanCounts_noNA$log2FC <= -2)
```

```
[1] 485
```

Q 10.Do you trust these results? Why or why not?

Not particularly, I would trust them more with significance values

DESeq analysis

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

```
library(DESeq2)
```

DESeq wants 3 things for analysis, countData, colData and design.

```
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 the DESeq package to run analysis is called `DESeq()`

```
# dds, has a slot in it for results
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 of this DESeq object with the function `results()`

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

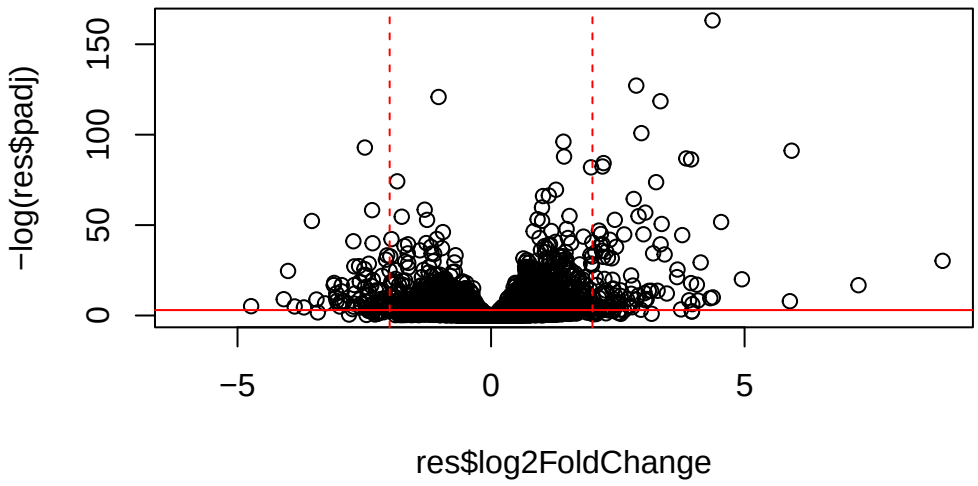
```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 6 columns
                  baseMean log2FoldChange     lfcSE      stat    pvalue
                 <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003 747.194195 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

```
# baseMean is the mean across all 8 columns
# lof2FC across treated/control
# lfcSE
#stat
#pvalue from statistical test
#padj is the adjusted pvalue, which is what we will use bc 0.05 of 36000 is still a very big
```

Volcano plot

This is a plot of log2FC v adjusted p-value

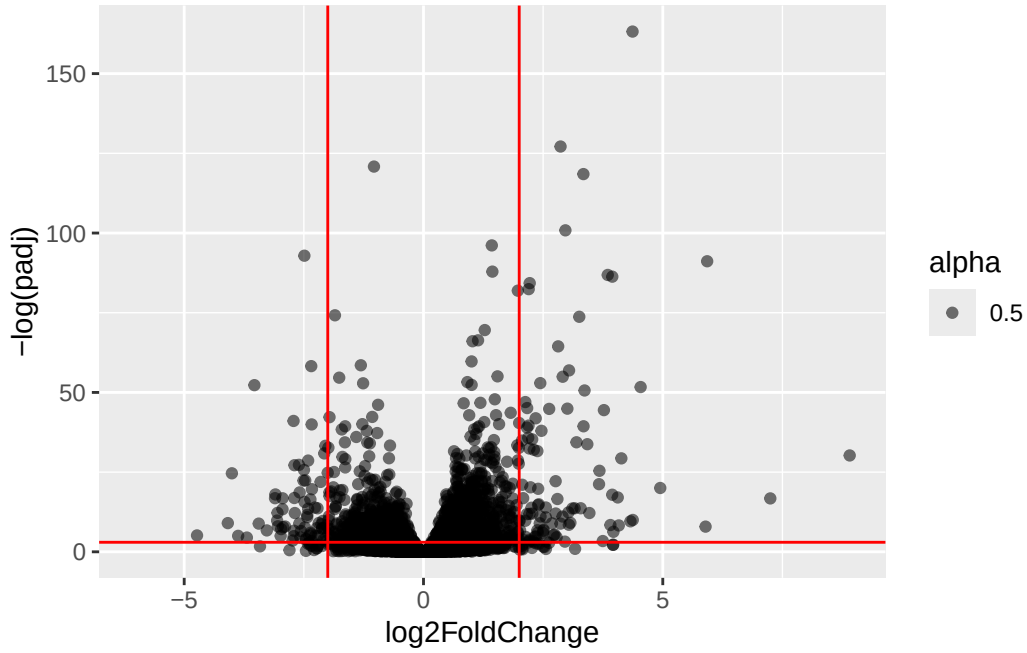
```
# have to take the log because all the values you care about are very compressed. You do -log
plot(res$log2FoldChange, -log(res$padj)) +
  abline(v=c(-2,2), col="red", lt="dashed") +
  abline(h=-log(0.05), col="red")
```




```
integer(0)
```

```
ggplot(res, aes(x=log2FoldChange, y=-log(padj), alpha=0.5)) +
  geom_point() +
  geom_hline(yintercept=-log(0.05), colour="red") +
  geom_vline(xintercept = c(2,-2), colour="red")
```

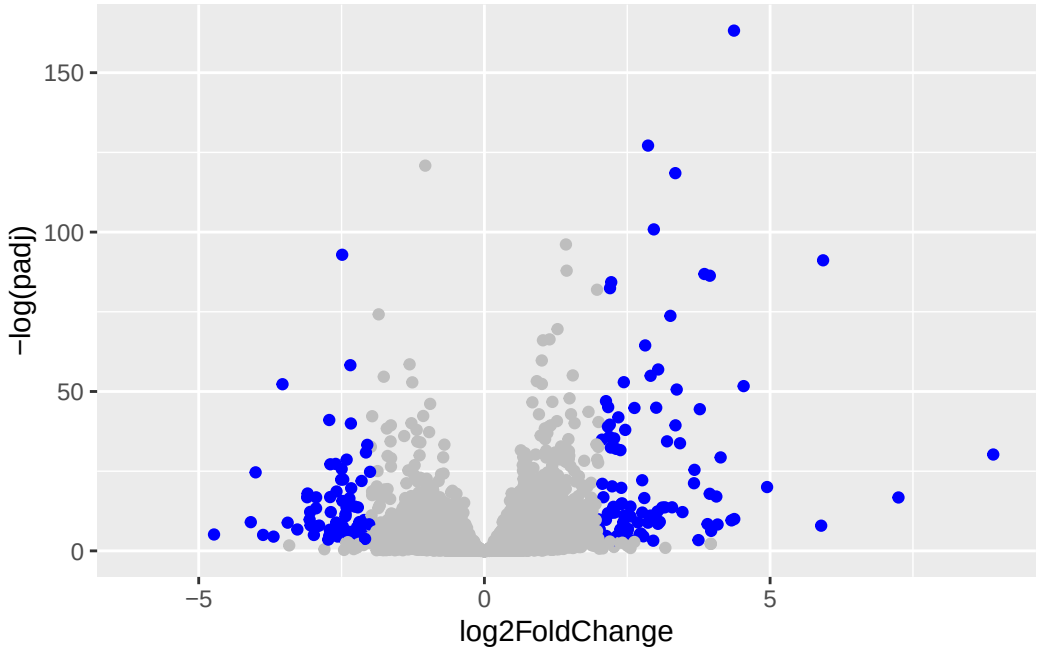
```
Warning: Removed 23549 rows containing missing values or values outside the scale range
(`geom_point()`).
```




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

ggplot(res, aes(x=log2FoldChange, y=-log(padj))) +
  geom_point(col=mycols)
```

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



Save our results

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

We skipped the PCA section to be resumed Wednesday.

Pathway 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)

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

```
[1] -0.35070296          NA  0.20610728  0.02452701 -0.14714263 -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 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/elizabethgilmore/Desktop/UCSD_Salk/BGGN213/BGGN213_Rsectio

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

```
# A different PDF based output of the same data
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/elizabethgilmore/Desktop/UCSD_Salk/BGGN213/BGGN213_Rsectio

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

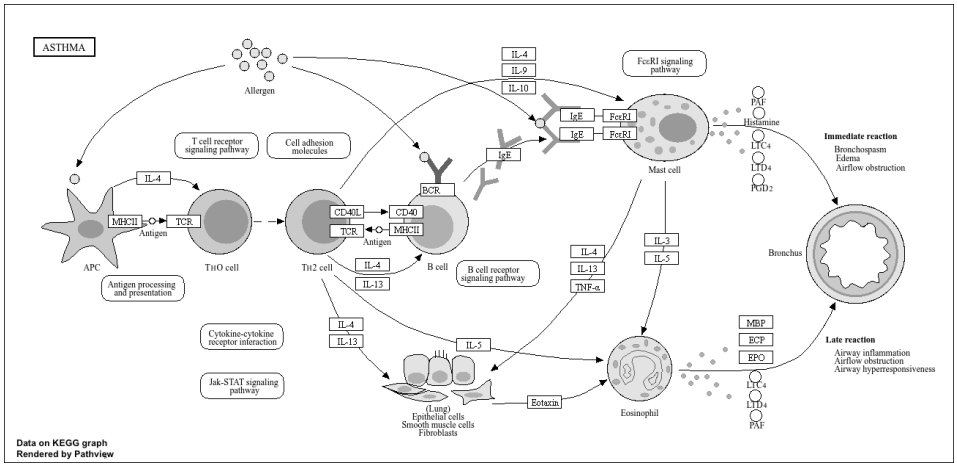


Figure 1: KEGG pathway hsa05310 (Asthma) generated by Pathview