

class13

```
[1] 4
```

```
#install.packages("BiocManager")
#BiocManager::install()
# For this class we will need DESeq2:
#BiocManager::install("DESeq2")

library(BiocManager)
library(DESeq2)
```

```
Loading required package: S4Vectors
```

```
Loading required package: stats4
```

```
Loading required package: BiocGenerics
```

```
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, intersect, is.unsorted, lapply, Map, mapply,
match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
table, tapply, union, 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: GenomeInfoDb
```

```
Loading required package: SummarizedExperiment
```

```
Warning: package 'SummarizedExperiment' was built under R version 4.3.2
```

```
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
```

Importing countData and colData

```
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

ENSG000000000938	0	0	1	0	2
	SRR1039517	SRR1039520	SRR1039521		
ENSG000000000003	1097	806	604		
ENSG000000000005	0	0	0		
ENSG000000000419	781	417	509		
ENSG000000000457	447	330	324		
ENSG000000000460	94	102	74		
ENSG000000000938	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

```
dim(metadata)
```

[1] 8 4

Q1. How many genes are in this dataset?

8 > Q2. How many ‘control’ cell lines do we have?

4

Compare the control to the treated columns. Steps:

- Step 1: ID and extract the “control” columns
- Step 2: calculate the mean value per gene for all these “control” columns
- Step 3: do the same for treated
- Step 4: compare the ‘control.mean’ and ‘treated.mean’ values

Step1:

```
control inds <- metadata$dex=="control"

metadata[control inds,]
```

```
    id      dex celltype     geo_id
1 SRR1039508 control   N61311 GSM1275862
3 SRR1039512 control   N052611 GSM1275866
5 SRR1039516 control   N080611 GSM1275870
7 SRR1039520 control   N061011 GSM1275874
```

Ensembl gene ID: ENSG000000000003 Metadata sheet: SRR1039508

```
control.mean <- rowMeans(counts[,control.ind])
head(control.mean)
```

```
ENSG000000000003 ENSG000000000005 ENSG000000000419 ENSG000000000457 ENSG000000000460
         900.75          0.00        520.50        339.75        97.25
ENSG000000000938
         0.75
```

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

```
ENSG000000000003 ENSG000000000005 ENSG000000000419 ENSG000000000457 ENSG000000000460
         658.00          0.00        546.00        316.50        78.75
ENSG000000000938
         0.00
```

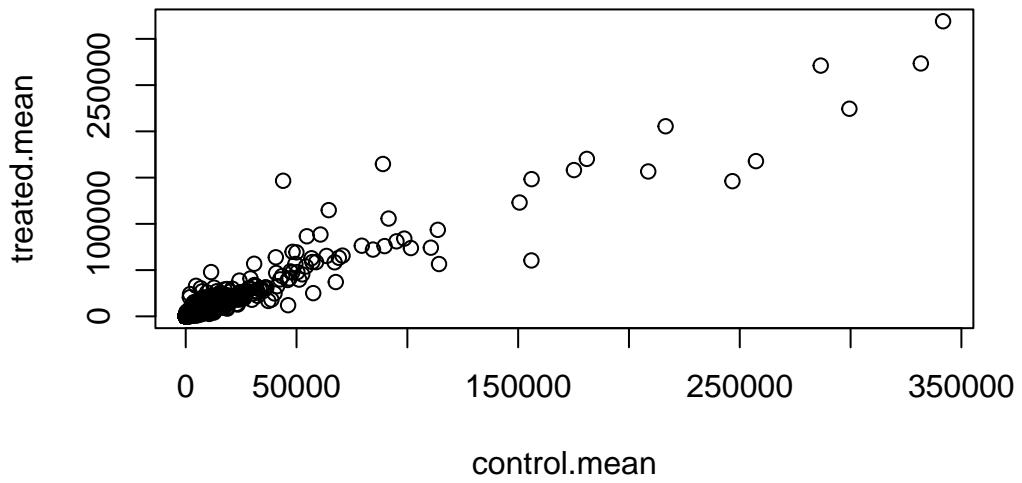
We will combine our meancount data for bookkeeping purposes.

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

Lets see what these count values look like....

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.

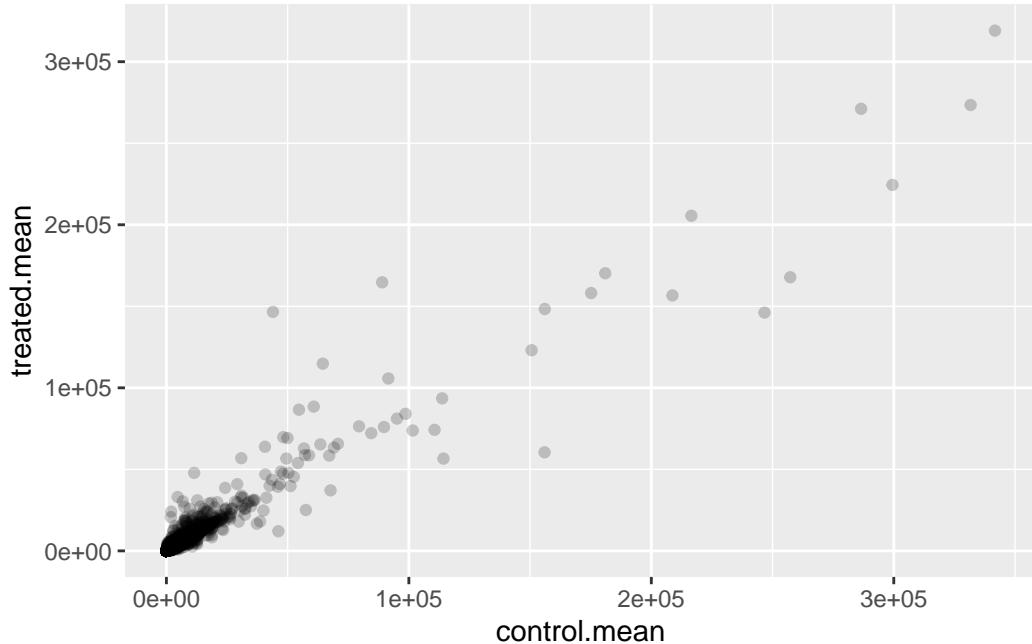
```
plot(meancounts)
```



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?

```
library(ggplot2)

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



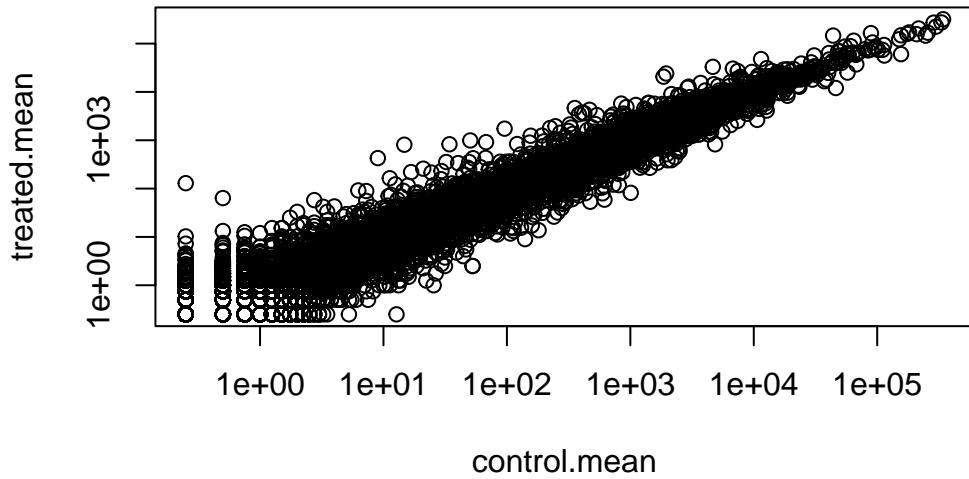
Q6. Try plotting both axes on a log scale. What is the argument to `plot()` that allows you to do this?

```
log="xy" or scale_x_continuous(trans="log2")/scale_y_continuous(trans="log2")
```

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

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

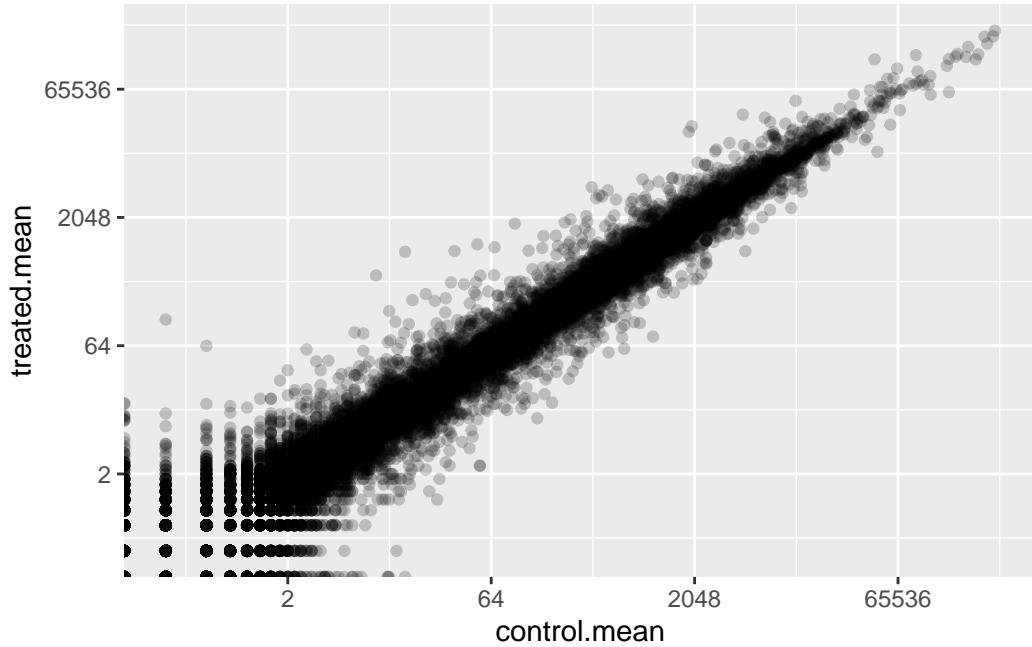
```
Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted
from logarithmic plot
```



```
#in ggplot
ggplot(meancounts) +
  aes(control.mean, treated.mean) +
  geom_point(alpha=0.2) +
  scale_x_continuous(trans="log2") +
  scale_y_continuous(trans="log2")
```

Warning: Transformation introduced infinite values in continuous x-axis

Warning: Transformation introduced infinite values in continuous y-axis



Logs are super useful when we have such skewed data they are also handy when we are most interested in

```
#treated/control
10/10
```

```
[1] 1
```

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

```
[1] 1
```

add log2(Fold-change) values to our results table.

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

head(meancounts)
```

	control.mean	treated.mean	log2fc
ENSG000000000003	900.75	658.00	-0.45303916
ENSG000000000005	0.00	0.00	NaN
ENSG000000000419	520.50	546.00	0.06900279
ENSG000000000457	339.75	316.50	-0.10226805
ENSG000000000460	97.25	78.75	-0.30441833
ENSG000000000938	0.75	0.00	-Inf

There are NaN and -Inf results. The NaN is returned when you divide by zero and try to take the log. The -Inf is returned when you try to take the log of zero.

Need to exclude any genes with zero counts as we can't say anything about them anyway from this experiment and it causes me math pain. >_<

```
#what values in the first two cols are zero
to.rm.ind<-rowSums(meancounts[, 1:2] == 0) >0
mycounts<-meancounts[!to.rm.ind,]
```

```
which(c(T, F, T))
```

```
[1] 1 3
```

This is another way to do it:

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

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

	control.mean	treated.mean	log2fc
ENSG000000000003	900.75	658.00	-0.45303916
ENSG000000000419	520.50	546.00	0.06900279
ENSG000000000457	339.75	316.50	-0.10226805
ENSG000000000460	97.25	78.75	-0.30441833
ENSG000000000971	5219.00	6687.50	0.35769358
ENSG00000001036	2327.00	1785.75	-0.38194109

Q. How many genes do i have left?

```
nrow(mycounts)
```

```
[1] 21817
```

Q8. How many genes are “up regulated” ie. have a $\log_2(\text{fold-change})$ greater than +2?

```
sum(mycounts$log2fc > +2)
```

```
[1] 250
```

Q9. How many are “down regulated”

```
sum(mycounts$log2f < -2)
```

```
[1] 367
```

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

No I don’t trust because we don’t know if these are significant.

setting up for DESeq

Like many bioconductor analysis packages DESeq wants it’s input in a very particular way.

```
library(DESeq2)
citation("DESeq2")
```

To cite package ‘DESeq2’ in publications use:

Love, M.I., Huber, W., Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2 *Genome Biology* 15(12):550 (2014)

A BibTeX entry for LaTeX users is

```
@Article{,
  title = {Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2},
  author = {Michael I. Love and Wolfgang Huber and Simon Anders},
```

```
year = {2014},  
journal = {Genome Biology},  
doi = {10.1186/s13059-014-0550-8},  
volume = {15},  
issue = {12},  
pages = {550},  
}
```

```
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

To run DESeq analysis we call the main function from the package called ‘DESeq(dds)’

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

estimating size factors

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

final dispersion estimates

fitting model and testing

To get the results out of this ‘dds’ object we can use the DESeq ‘results()’ function.

```
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>
ENSG000000000003 747.194195 -0.3507030 0.168246 -2.084470 0.0371175
ENSG000000000005 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>
ENSG000000000003 0.163035
ENSG000000000005  NA
ENSG00000000419 0.176032
ENSG00000000457 0.961694
ENSG00000000460 0.815849
ENSG00000000938  NA

```

A common summary visualization is called a volcano plot.

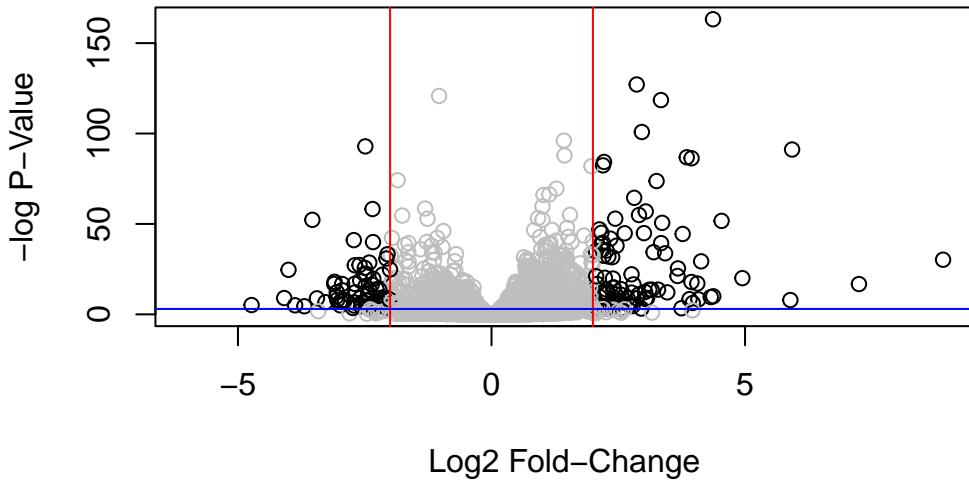
```

mycols <- rep("gray", nrow(res))
mycols[res$log2FoldChange > 2] <- "black"
mycols[res$log2FoldChange < -2] <- "black"
mycols[res$padj > 0.05] <- "gray"

plot(res$log2FoldChange, -log(res$padj), col=mycols,
      xlab="Log2 Fold-Change",
      ylab="-log P-Value")

abline(v=c(2, -2), col="red")
abline(h=-log(0.05), col="blue")

```



```
#Save our results to date
```

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

we need to translate or “map” our ensemble ID’s into more understandable gene names and the identifiers that other useful databases use.

```
BiocManager::install("AnnotationDbi")
```

```
Bioconductor version 3.18 (BiocManager 1.30.22), R 4.3.1 (2023-06-16)
```

```
Warning: package(s) not installed when version(s) same as or greater than current; use
`force = TRUE` to re-install: 'AnnotationDbi'
```

```
Old packages: 'GenomeInfoDb', 'httr2', 'lifecycle', 'lme4', 'Matrix',
'MatrixModels', 'matrixStats', 'rlang', 'rprojroot', 'RSQLite', 'stringi',
'stringr'
```

```
BiocManager::install("org.Hs.eg.db")
```

```
Bioconductor version 3.18 (BiocManager 1.30.22), R 4.3.1 (2023-06-16)
```

```
Warning: package(s) not installed when version(s) same as or greater than current; use  
`force = TRUE` to re-install: 'org.Hs.eg.db'
```

```
Old packages: 'GenomeInfoDb', 'httr2', 'lifecycle', 'lme4', 'Matrix',  
'MatrixModels', 'matrixStats', 'rlang', 'rprojroot', 'RSQLite', 'stringi',  
'stringr'
```

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

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

```
[1] "ACNUM"          "ALIAS"           "ENSEMBL"         "ENSEMLPROT"      "ENSEMLTRANS"  
[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>
ENSG000000000003 747.194195 -0.3507030 0.168246 -2.084470 0.0371175
ENSG000000000005 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>
ENSG000000000003 0.163035   TSPAN6
ENSG000000000005  NA        TNMD
ENSG00000000419 0.176032   DPM1
ENSG00000000457 0.961694   SCYL3
ENSG00000000460 0.815849   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>
ENSG000000000003	747.194195	-0.3507030	0.168246	-2.084470	0.0371175
ENSG000000000005	0.000000	NA	NA	NA	NA
ENSG000000000419	520.134160	0.2061078	0.101059	2.039475	0.0414026
ENSG000000000457	322.664844	0.0245269	0.145145	0.168982	0.8658106
ENSG000000000460	87.682625	-0.1471420	0.257007	-0.572521	0.5669691
ENSG000000000938	0.319167	-1.7322890	3.493601	-0.495846	0.6200029
	padj	symbol	entrez	uniprot	
	<numeric>	<character>	<character>	<character>	
ENSG000000000003	0.163035	TSPAN6	7105	AOA024RCI0	
ENSG000000000005	NA	TNMD	64102	Q9H2S6	
ENSG000000000419	0.176032	DPM1	8813	060762	
ENSG000000000457	0.961694	SCYL3	57147	Q8IZE3	
ENSG000000000460	0.815849	FIRRM	55732	AOA024R922	
ENSG000000000938	NA	FGR	2268	P09769	
	genename				
	<character>				
ENSG000000000003	tetraspanin 6				
ENSG000000000005	tenomodulin				
ENSG000000000419	dolichyl-phosphate m..				
ENSG000000000457	SCY1 like pseudokina..				
ENSG000000000460	FIGNL1 interacting r..				
ENSG000000000938	FGR proto-oncogene, ..				

```
# Run in your R console (i.e. not your Rmarkdown doc!)
BiocManager::install( c("pathview", "gage", "gageData") )
```

```
Bioconductor version 3.18 (BiocManager 1.30.22), R 4.3.1 (2023-06-16)
```

```
Warning: package(s) not installed when version(s) same as or greater than current; use
`force = TRUE` to re-install: 'pathview' 'gage' 'gageData'
```

```
Old packages: 'GenomeInfoDb', 'httr2', 'lifecycle', 'lme4', 'Matrix',
'MatrixModels', 'matrixStats', 'rlang', 'rprojroot', 'RSQLite', 'stringi',
'stringr'
```

```
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)  

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
```

lets have a look at one of these pathways

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

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

```
Info: Working in directory /Users/e ebonyargaez/Desktop/FALL 2023/BGGN213_Bioinformatics/
```

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

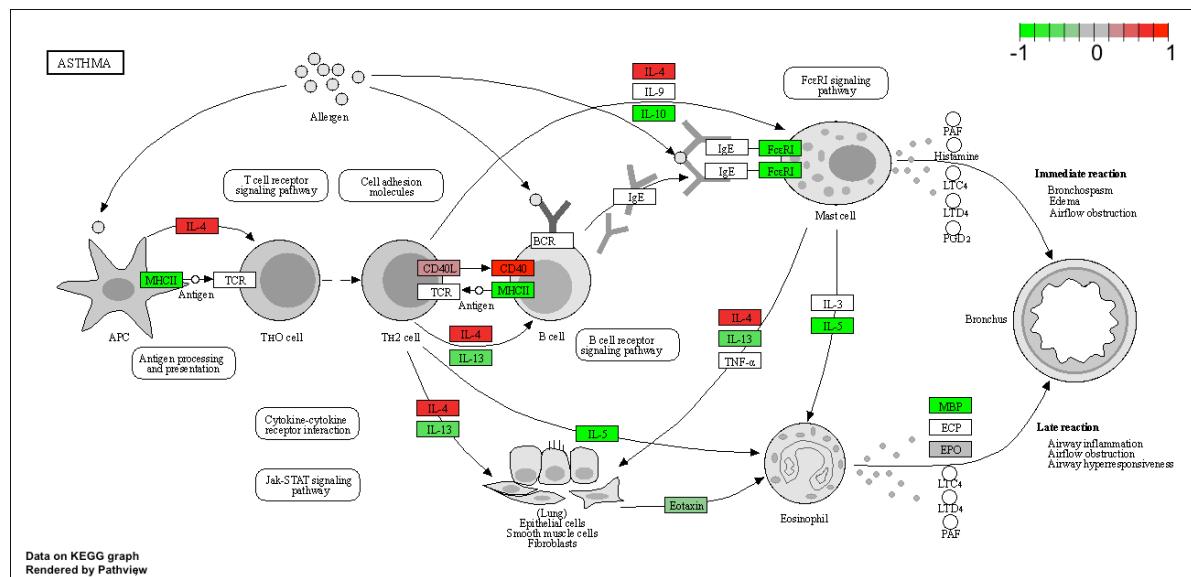


Figure 1: The Asthma pathway from KEGG