

Class 12

Olivia Solorzano Alvarez

```
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
```

```
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
```

Import their data

We need two things for this analysis:

- **counts** (counts for every transcript/gene in each experiment)
- **metadata** (metadata that describes the experimental setup)

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

	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516
ENSG000000000003	723	486	904	445	1170
ENSG000000000005	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		

```

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)
```

	<code>id</code>	<code>dex</code>	<code>celltype</code>	<code>geo_id</code>
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 ‘control’ cell lines do we have?

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

```

control treated
4          4

```

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

```
[1] 4
```

Step 1 Calculate the mean of the control samples (i.e. columns in `counts`)

(a) We need to find which columns in `counts` are “control” samples.

- look in the metadata (a.k.a. `metadata`), `$dex` column

```
control.ind <- metadata$dex == "control"
```

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

```
control.counts <- counts[,control.ind]
```

- (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
ENSG000000000938
      0.75
```

Q4.

- Step 2. Calculate the mean of the treated samples...

```
# Extract treated data columns (displayed as logicals)
treated.ind <- metadata$dex == "treated"
```

```
# Count logicals that display TRUE for treated"
treated.counts <- counts[,treated.ind]
```

```
# Calculate mean of each gene (row) that is classified as treated
treated.means <- rowMeans( treated.counts )
head(treated.means)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
      658.00          0.00        546.00        316.50        78.75
ENSG000000000938
      0.00
```

We now have control and treated mean count values. For ease of book-keeping I will combine these vectors into a data.frame called **meancounts**

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

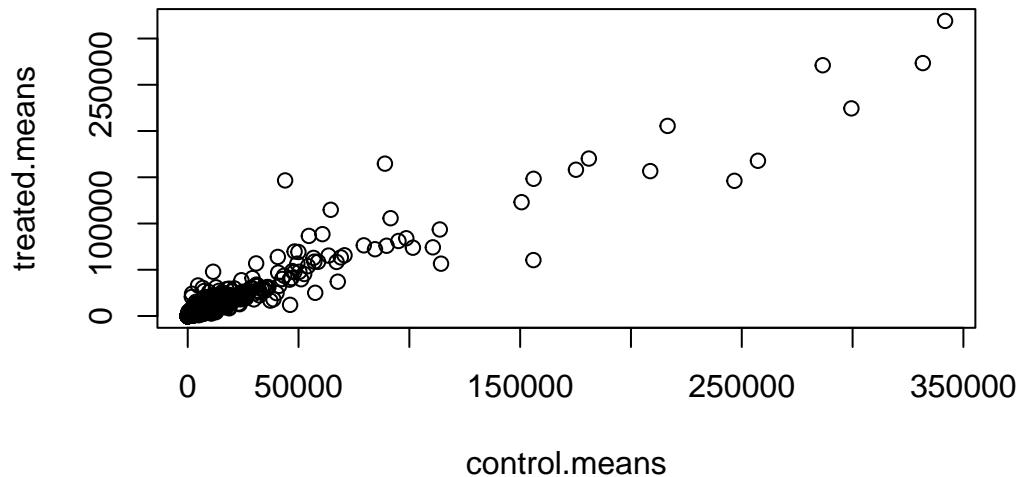
	control.means	treated.means
ENSG000000000003	900.75	658.00
ENSG000000000005	0.00	0.00
ENSG00000000419	520.50	546.00
ENSG00000000457	339.75	316.50
ENSG00000000460	97.25	78.75
ENSG00000000938	0.75	0.00

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.

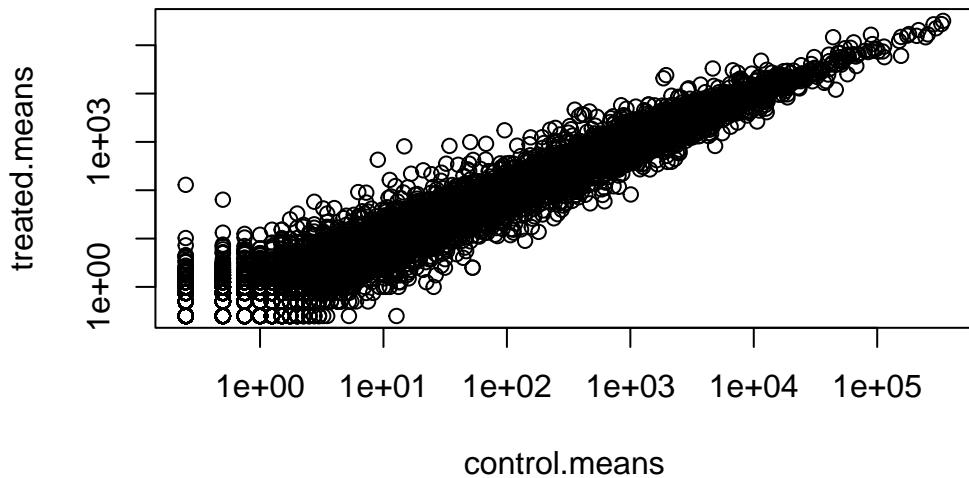
```
plot(meancounts)
```



```
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
```



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

We must often use log2 as our transform as the math is easier to interpret than log10 or others.

If we have no change - i.e. same values in control and treated we will have a log2 value of zero

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

```
[1] 0
```

If I have double the amount i.e. 20 compared to 10 for example I will have a log2 value of 1.

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

```
[1] 1
```

If I have half the amount I will have a log2 fold-change of -1

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

```
[1] -1
```

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

```
[1] 2
```

```
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

Q. How many genes are up regulated at the common threshold of +2 log2FC

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

```
[1] 1910
```

Hold on what about the stats! Yes, these are big changes, but are these changes significant!

To do this properly we will turn to the DESeq2 package.

```
#DESeq2 analysis
```

```
library(DESeq2)
```

To use DESeq we need our inout contData and colData in a specific format that DESeq wants:

```
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 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 the results out of this `dds` object we can use the `results()` function from the 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
```

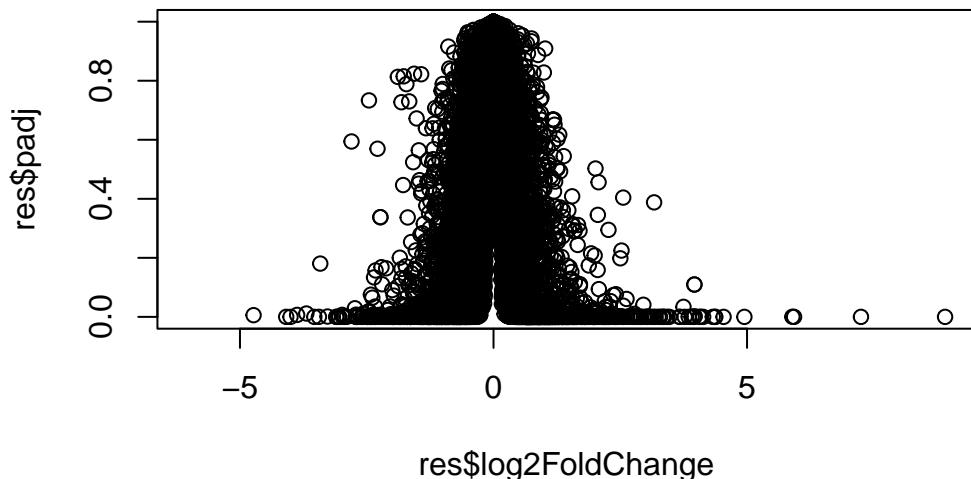
```

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
<numeric>
ENSG000000000003  0.163035
ENSG000000000005  NA
ENSG000000000419  0.176032
ENSG000000000457  0.961694
ENSG000000000460  0.815849
ENSG000000000938  NA

```

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

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

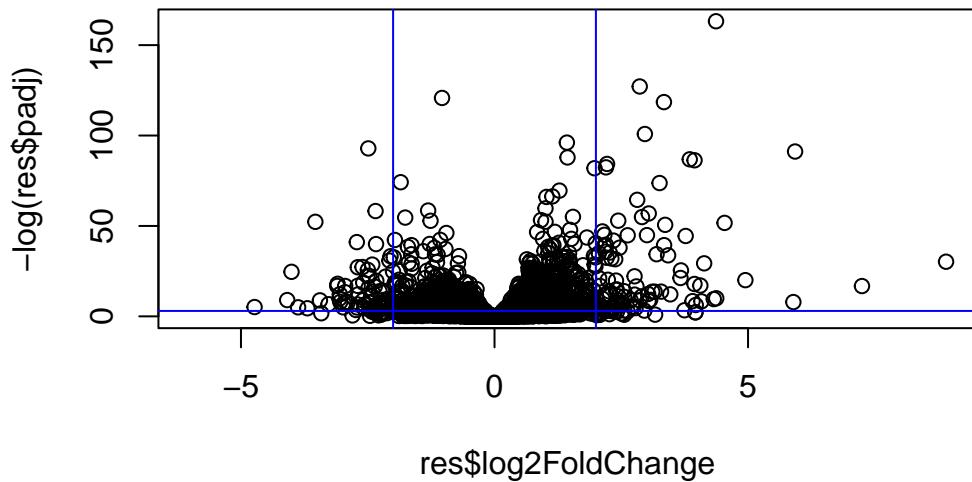


It is the low P-values that we care about and these are lost in the skewed plot above. Let's take the log of the \$padj values for our plot

```

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

```



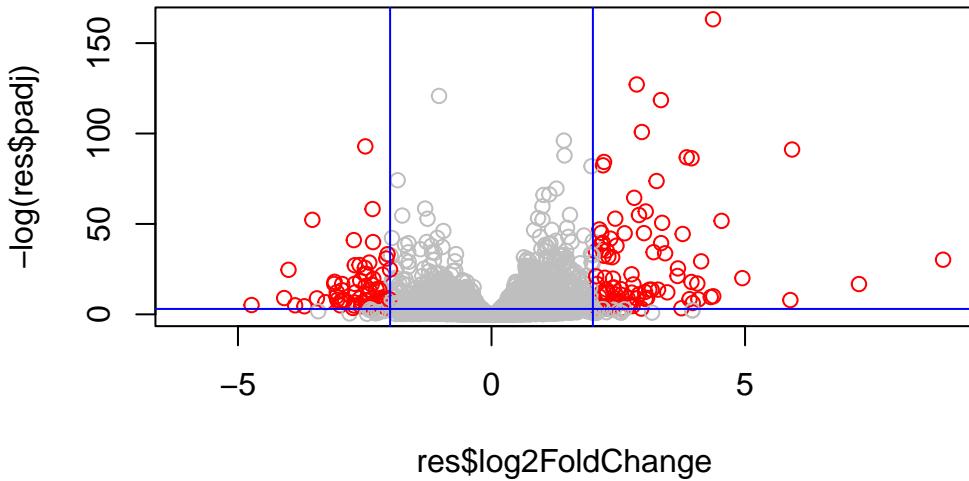
Finally we can make a color vector to use in the plot to better highlight the genes we care about.

```

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")

```



Done for today :-)

Still to do: 1. Add annotation (including gene symbols, etrezid, and genoname) 2. Save results as CSV file. 3. Do same Pathway Analysis (KEGG and GO)

Adding annotation data

We can use the AnnotationDbi package to add annotation data such as gene identifiers from different sources to our results object.

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

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

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

```
[1] "ACCCNUM"      "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"
```

My IDs are stored as the rownames of `res`

```
head( rownames(res) )

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

res$symbol <- mapIds(org.Hs.eg.db,
                      keys=rownames(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 -0.3507030  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      symbol
  <numeric> <character>
ENSG00000000003  0.163035    TSPAN6
ENSG00000000005      NA        TNMD
```

```
ENSG00000000419 0.176032      DPM1
ENSG00000000457 0.961694      SCYL3
ENSG00000000460 0.815849      FIRRM
ENSG00000000938 NA             FGR
```

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

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

Save our results to date

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

Pathway analysis

We can use the KEGG database of biological pathways to get some more insight into our differentially expressed genes and the kinds of biology that are involved in.

```
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>).

```
#####
```

Look at the first two KEGG pathways

```
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"
```

Make a new vector of fold-change

```
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

```

x <- 1:3
x

[1] 1 2 3

names(x) <- c("chandra", "lisa", "xinqiu")
x

chandra     lisa    xinqiu
1           2         3

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)

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

attributes(keggres$less)

$dim
[1] 229    6

$dimnames
```

```

$dimnames[[1]]
[1] "hsa05332 Graft-versus-host disease"
[2] "hsa04940 Type I diabetes mellitus"
[3] "hsa05310 Asthma"
[4] "hsa04672 Intestinal immune network for IgA production"
[5] "hsa05330 Allograft rejection"
[6] "hsa04340 Hedgehog signaling pathway"
[7] "hsa05150 Staphylococcus aureus infection"
[8] "hsa04916 Melanogenesis"
[9] "hsa04972 Pancreatic secretion"
[10] "hsa05140 Leishmaniasis"
[11] "hsa04612 Antigen processing and presentation"
[12] "hsa05217 Basal cell carcinoma"
[13] "hsa04640 Hematopoietic cell lineage"
[14] "hsa05144 Malaria"
[15] "hsa05320 Autoimmune thyroid disease"
[16] "hsa05143 African trypanosomiasis"
[17] "hsa05323 Rheumatoid arthritis"
[18] "hsa04380 Osteoclast differentiation"
[19] "hsa04971 Gastric acid secretion"
[20] "hsa00565 Ether lipid metabolism"
[21] "hsa05142 Chagas disease (American trypanosomiasis)"
[22] "hsa04020 Calcium signaling pathway"
[23] "hsa04514 Cell adhesion molecules (CAMs)"
[24] "hsa04742 Taste transduction"
[25] "hsa00533 Glycosaminoglycan biosynthesis - keratan sulfate"
[26] "hsa05340 Primary immunodeficiency"
[27] "hsa04740 Olfactory transduction"
[28] "hsa03030 DNA replication"
[29] "hsa00561 Glycerolipid metabolism"
[30] "hsa00970 Aminoacyl-tRNA biosynthesis"
[31] "hsa04912 GnRH signaling pathway"
[32] "hsa03410 Base excision repair"
[33] "hsa04540 Gap junction"
[34] "hsa00603 Glycosphingolipid biosynthesis - globo series"
[35] "hsa04115 p53 signaling pathway"
[36] "hsa04080 Neuroactive ligand-receptor interaction"
[37] "hsa00740 Riboflavin metabolism"
[38] "hsa04974 Protein digestion and absorption"
[39] "hsa00591 Linoleic acid metabolism"
[40] "hsa03430 Mismatch repair"
[41] "hsa00230 Purine metabolism"
[42] "hsa05160 Hepatitis C"

```

- [43] "hsa04610 Complement and coagulation cascades"
- [44] "hsa04060 Cytokine-cytokine receptor interaction"
- [45] "hsa04720 Long-term potentiation"
- [46] "hsa03420 Nucleotide excision repair"
- [47] "hsa04744 Phototransduction"
- [48] "hsa00920 Sulfur metabolism"
- [49] "hsa04623 Cytosolic DNA-sensing pathway"
- [50] "hsa04975 Fat digestion and absorption"
- [51] "hsa05211 Renal cell carcinoma"
- [52] "hsa03440 Homologous recombination"
- [53] "hsa04976 Bile secretion"
- [54] "hsa04962 Vasopressin-regulated water reabsorption"
- [55] "hsa04360 Axon guidance"
- [56] "hsa04620 Toll-like receptor signaling pathway"
- [57] "hsa00120 Primary bile acid biosynthesis"
- [58] "hsa04110 Cell cycle"
- [59] "hsa04320 Dorso-ventral axis formation"
- [60] "hsa04310 Wnt signaling pathway"
- [61] "hsa05145 Toxoplasmosis"
- [62] "hsa05212 Pancreatic cancer"
- [63] "hsa00592 alpha-Linolenic acid metabolism"
- [64] "hsa04210 Apoptosis"
- [65] "hsa00511 Other glycan degradation"
- [66] "hsa04664 Fc epsilon RI signaling pathway"
- [67] "hsa04114 Oocyte meiosis"
- [68] "hsa04730 Long-term depression"
- [69] "hsa00630 Glyoxylate and dicarboxylate metabolism"
- [70] "hsa04914 Progesterone-mediated oocyte maturation"
- [71] "hsa05130 Pathogenic Escherichia coli infection"
- [72] "hsa04145 Phagosome"
- [73] "hsa04012 ErbB signaling pathway"
- [74] "hsa04330 Notch signaling pathway"
- [75] "hsa04970 Salivary secretion"
- [76] "hsa04144 Endocytosis"
- [77] "hsa03013 RNA transport"
- [78] "hsa04650 Natural killer cell mediated cytotoxicity"
- [79] "hsa04666 Fc gamma R-mediated phagocytosis"
- [80] "hsa05216 Thyroid cancer"
- [81] "hsa04622 RIG-I-like receptor signaling pathway"
- [82] "hsa00900 Terpenoid backbone biosynthesis"
- [83] "hsa04660 T cell receptor signaling pathway"
- [84] "hsa00514 Other types of O-glycan biosynthesis"
- [85] "hsa05219 Bladder cancer"

- [86] "hsa05200 Pathways in cancer"
- [87] "hsa05322 Systemic lupus erythematosus"
- [88] "hsa05215 Prostate cancer"
- [89] "hsa05218 Melanoma"
- [90] "hsa00240 Pyrimidine metabolism"
- [91] "hsa04960 Aldosterone-regulated sodium reabsorption"
- [92] "hsa00260 Glycine, serine and threonine metabolism"
- [93] "hsa03010 Ribosome"
- [94] "hsa05214 Glioma"
- [95] "hsa04670 Leukocyte transendothelial migration"
- [96] "hsa04621 NOD-like receptor signaling pathway"
- [97] "hsa00340 Histidine metabolism"
- [98] "hsa04662 B cell receptor signaling pathway"
- [99] "hsa00860 Porphyrin and chlorophyll metabolism"
- [100] "hsa03015 mRNA surveillance pathway"
- [101] "hsa04964 Proximal tubule bicarbonate reclamation"
- [102] "hsa04062 Chemokine signaling pathway"
- [103] "hsa04950 Maturity onset diabetes of the young"
- [104] "hsa02010 ABC transporters"
- [105] "hsa05016 Huntington's disease"
- [106] "hsa04973 Carbohydrate digestion and absorption"
- [107] "hsa04930 Type II diabetes mellitus"
- [108] "hsa04130 SNARE interactions in vesicular transport"
- [109] "hsa00512 Mucin type O-Glycan biosynthesis"
- [110] "hsa04120 Ubiquitin mediated proteolysis"
- [111] "hsa03020 RNA polymerase"
- [112] "hsa03060 Protein export"
- [113] "hsa04010 MAPK signaling pathway"
- [114] "hsa00510 N-Glycan biosynthesis"
- [115] "hsa04146 Peroxisome"
- [116] "hsa00030 Pentose phosphate pathway"
- [117] "hsa00670 One carbon pool by folate"
- [118] "hsa04614 Renin-angiotensin system"
- [119] "hsa00310 Lysine degradation"
- [120] "hsa03040 Spliceosome"
- [121] "hsa00360 Phenylalanine metabolism"
- [122] "hsa03008 Ribosome biogenesis in eukaryotes"
- [123] "hsa00290 Valine, leucine and isoleucine biosynthesis"
- [124] "hsa05213 Endometrial cancer"
- [125] "hsa03018 RNA degradation"
- [126] "hsa05223 Non-small cell lung cancer"
- [127] "hsa04512 ECM-receptor interaction"
- [128] "hsa04710 Circadian rhythm - mammal"

- [129] "hsa00983 Drug metabolism - other enzymes"
- [130] "hsa03450 Non-homologous end-joining"
- [131] "hsa00051 Fructose and mannose metabolism"
- [132] "hsa05020 Prion diseases"
- [133] "hsa04630 Jak-STAT signaling pathway"
- [134] "hsa04141 Protein processing in endoplasmic reticulum"
- [135] "hsa04370 VEGF signaling pathway"
- [136] "hsa05010 Alzheimer's disease"
- [137] "hsa00600 Sphingolipid metabolism"
- [138] "hsa05210 Colorectal cancer"
- [139] "hsa00650 Butanoate metabolism"
- [140] "hsa05014 Amyotrophic lateral sclerosis (ALS)"
- [141] "hsa00770 Pantothenate and CoA biosynthesis"
- [142] "hsa04150 mTOR signaling pathway"
- [143] "hsa05222 Small cell lung cancer"
- [144] "hsa00564 Glycerophospholipid metabolism"
- [145] "hsa00140 Steroid hormone biosynthesis"
- [146] "hsa04977 Vitamin digestion and absorption"
- [147] "hsa01040 Biosynthesis of unsaturated fatty acids"
- [148] "hsa03050 Proteasome"
- [149] "hsa05220 Chronic myeloid leukemia"
- [150] "hsa00563 Glycosylphosphatidylinositol(GPI)-anchor biosynthesis"
- [151] "hsa00250 Alanine, aspartate and glutamate metabolism"
- [152] "hsa04142 Lysosome"
- [153] "hsa03320 PPAR signaling pathway"
- [154] "hsa00604 Glycosphingolipid biosynthesis - ganglio series"
- [155] "hsa05414 Dilated cardiomyopathy"
- [156] "hsa00910 Nitrogen metabolism"
- [157] "hsa04722 Neurotrophin signaling pathway"
- [158] "hsa00980 Metabolism of xenobiotics by cytochrome P450"
- [159] "hsa04350 TGF-beta signaling pathway"
- [160] "hsa00430 Taurine and hypotaurine metabolism"
- [161] "hsa00410 beta-Alanine metabolism"
- [162] "hsa00100 Steroid biosynthesis"
- [163] "hsa00450 Selenocompound metabolism"
- [164] "hsa04270 Vascular smooth muscle contraction"
- [165] "hsa00520 Amino sugar and nucleotide sugar metabolism"
- [166] "hsa00052 Galactose metabolism"
- [167] "hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series"
- [168] "hsa00350 Tyrosine metabolism"
- [169] "hsa05100 Bacterial invasion of epithelial cells"
- [170] "hsa04070 Phosphatidylinositol signaling system"
- [171] "hsa04810 Regulation of actin cytoskeleton"

- [172] "hsa00562 Inositol phosphate metabolism"
- [173] "hsa00620 Pyruvate metabolism"
- [174] "hsa04122 Sulfur relay system"
- [175] "hsa00270 Cysteine and methionine metabolism"
- [176] "hsa05146 Amoebiasis"
- [177] "hsa04520 Adherens junction"
- [178] "hsa00590 Arachidonic acid metabolism"
- [179] "hsa00531 Glycosaminoglycan degradation"
- [180] "hsa05412 Arrhythmogenic right ventricular cardiomyopathy (ARVC)"
- [181] "hsa04966 Collecting duct acid secretion"
- [182] "hsa05110 Vibrio cholerae infection"
- [183] "hsa03022 Basal transcription factors"
- [184] "hsa00020 Citrate cycle (TCA cycle)"
- [185] "hsa00982 Drug metabolism - cytochrome P450"
- [186] "hsa04530 Tight junction"
- [187] "hsa00190 Oxidative phosphorylation"
- [188] "hsa04140 Regulation of autophagy"
- [189] "hsa00830 Retinol metabolism"
- [190] "hsa05012 Parkinson's disease"
- [191] "hsa00280 Valine, leucine and isoleucine degradation"
- [192] "hsa05410 Hypertrophic cardiomyopathy (HCM)"
- [193] "hsa05416 Viral myocarditis"
- [194] "hsa00760 Nicotinate and nicotinamide metabolism"
- [195] "hsa00480 Glutathione metabolism"
- [196] "hsa00640 Propanoate metabolism"
- [197] "hsa05221 Acute myeloid leukemia"
- [198] "hsa00053 Ascorbate and aldarate metabolism"
- [199] "hsa05120 Epithelial cell signaling in Helicobacter pylori infection"
- [200] "hsa00380 Tryptophan metabolism"
- [201] "hsa00010 Glycolysis / Gluconeogenesis"
- [202] "hsa04260 Cardiac muscle contraction"
- [203] "hsa00040 Pentose and glucuronate interconversions"
- [204] "hsa00534 Glycosaminoglycan biosynthesis - heparan sulfate"
- [205] "hsa05131 Shigellosis"
- [206] "hsa00071 Fatty acid metabolism"
- [207] "hsa00532 Glycosaminoglycan biosynthesis - chondroitin sulfate"
- [208] "hsa00790 Folate biosynthesis"
- [209] "hsa04920 Adipocytokine signaling pathway"
- [210] "hsa04510 Focal adhesion"
- [211] "hsa04910 Insulin signaling pathway"
- [212] "hsa00330 Arginine and proline metabolism"
- [213] "hsa00500 Starch and sucrose metabolism"
- [214] "hsa00232 Caffeine metabolism"

```
[215] "hsa01100 Metabolic pathways"
[216] "hsa00300 Lysine biosynthesis"
[217] "hsa00130 Ubiquinone and other terpenoid-quinone biosynthesis"
[218] "hsa00062 Fatty acid elongation in mitochondria"
[219] "hsa00785 Lipoic acid metabolism"
[220] "hsa00460 Cyanoamino acid metabolism"
[221] "hsa00472 D-Arginine and D-ornithine metabolism"
[222] "hsa00061 Fatty acid biosynthesis"
[223] "hsa00400 Phenylalanine, tyrosine and tryptophan biosynthesis"
[224] "hsa00524 Butirosin and neomycin biosynthesis"
[225] "hsa00730 Thiamine metabolism"
[226] "hsa00471 D-Glutamine and D-glutamate metabolism"
[227] "hsa00750 Vitamin B6 metabolism"
[228] "hsa00780 Biotin metabolism"
[229] "hsa00072 Synthesis and degradation of ketone bodies"
```

```
$dimnames[[2]]
[1] "p.geomean" "stat.mean" "p.val"      "q.val"      "set.size"   "exp1"
```

Now I can use the **KEGG IDs** (“hsa5310” etc.) of these pathways from gage to view our genes mapped to these pathways.

```
pathview(gene.data=foldchanges, pathway.id="hsa04110")
```

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

```
Info: Working in directory /Users/oliveom/Desktop/UCSD/Spring 2023/BIMM 143/Class 12
```

```
Info: Writing image file hsa04110.pathview.png
```

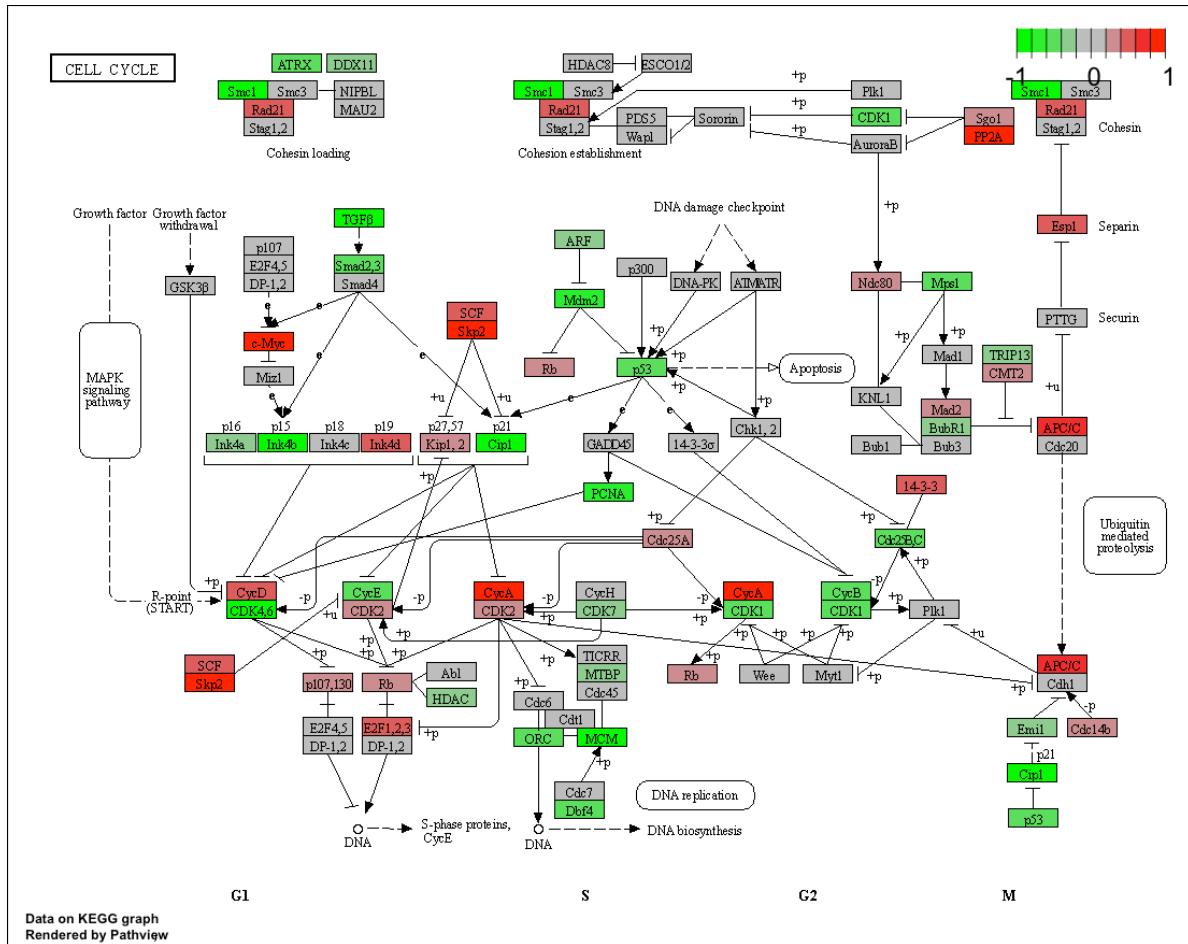


Figure 1: Asthma pathway from KEGG with our genes shown in color