

# Class 15

Shitian Li (PID: A13294481)

11/17/2021

## background

Today we examine a published RNA-seq experiment where airway smooth muscle cells were treated with

We need 2 things:/ -1: count data

-2: colData (the metadata that tells us about the design of the experiment)

```
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
## 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
```

Side note:

Let's check the correspondence of the metadata and count data setup.

```
all(metadata$id == colnames(counts))

## [1] TRUE
```

## Compare control to treated

First, we need to access all the control columns in our counts data.

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

Use these ids to access just the control columns f our counts data

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

```
##          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
```

Now do same for treat:

```
treat inds <- !control.ids
treat.ids <- metadata[treat inds,]$id
treat.mean <- rowMeans(counts[, treat.ids])
head(counts[, treat.ids])
```

```
##          SRR1039509 SRR1039513 SRR1039517 SRR1039521
## ENSG00000000003     486      445     1097      604
## ENSG00000000005      0         0         0         0
## ENSG00000000419    523      371      781      509
## ENSG00000000457    258      237      447      324
## ENSG00000000460     81       66      94       74
## ENSG00000000938      0         0         0         0
```

For book keeping, let's combine these into a new dataset.

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

There are 38694 rows/genes in this dataset.

```
nrow(counts)
```

```
## [1] 38694
```

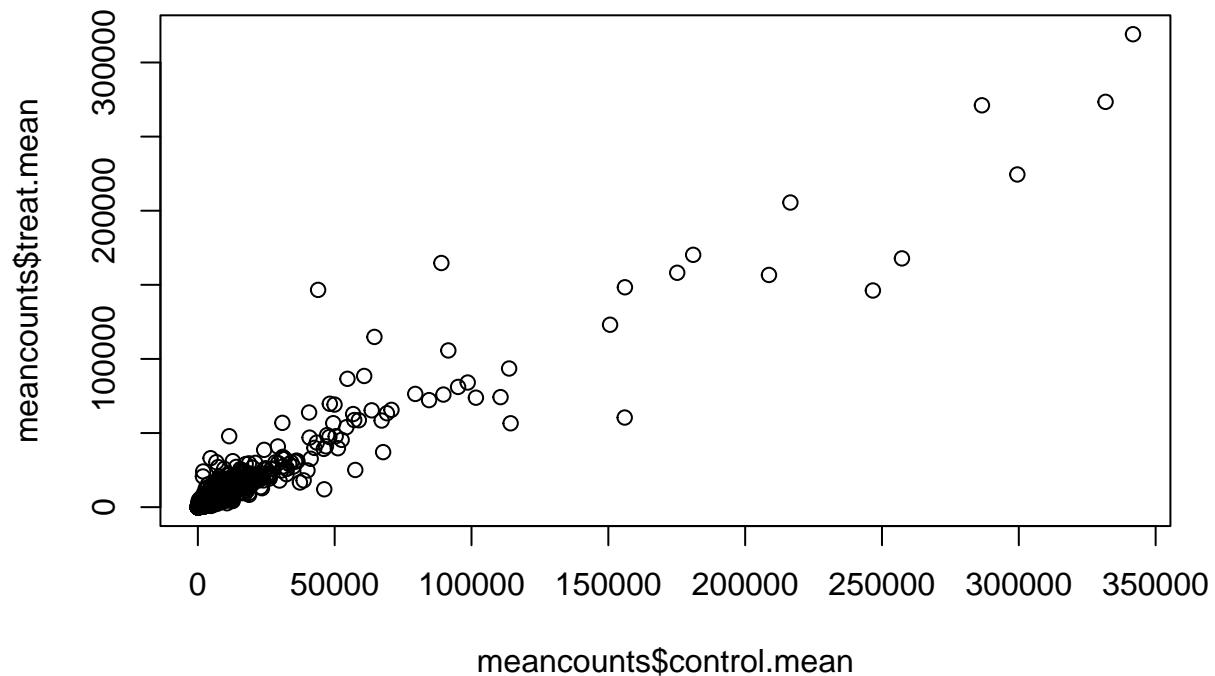
```
nrow(meancounts)
```

```
## [1] 38694
```

## Compare the control and treated:

Let's do a quick plot:

```
plot(meancounts$control.mean, meancounts$treat.mean)
```

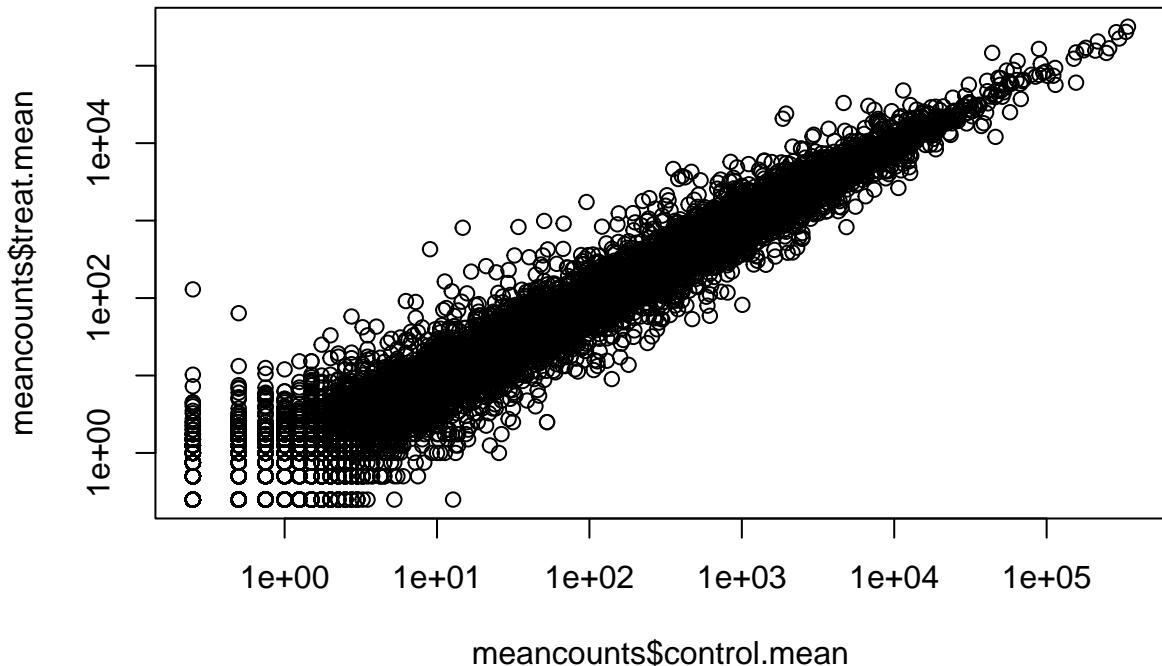


Let's put on a log scale:

```
plot(meancounts$control.mean, meancounts$treat.mean, 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 often use log transformations as they make life much nicer in this world...

```

log2(20/20)

## [1] 0

log2(40/20)

## [1] 1

log2(10/20)

## [1] -1

log2(80/20)

## [1] 2

meancounts$log2fc <- log2(meancounts[, "treat.mean"] / meancounts[, "control.mean"])
head(meancounts)

##          control.mean treat.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

```

Remove those with inf and NaN.

```

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

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

```

```

##                  control.mean treat.mean      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

```

We now have 21817 gene remaining:

```
nrow(mycounts)
```

```
## [1] 21817
```

How many of these genes are up regulated at the log2 fold-change threshold of +2 or greater?

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

```
## [1] 250
```

What percentage is this?

```
round((sum(mycounts$log2fc > +2) / nrow(mycounts)) * 100, 2)
```

```
## [1] 1.15
```

```
sum(mycounts$log2fc < -2)
```

```
## [1] 367
```

## DEseq2

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

```

##
## 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

dds

## class: DESeqDataSet
## dim: 38694 8
## metadata(1): version
## assays(1): counts
## rownames(38694): ENSG00000000003 ENSG00000000005 ... ENSG00000283120
##   ENSG00000283123
## rowData names(0):
## colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521
## colData names(4): id dex celltype geo_id

```

Run the DESeq analysis:

```

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:

```

res <- results(dds)
res

## log2 fold change (MLE): dex treated vs control
## Wald test p-value: dex treated vs control
## DataFrame with 38694 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat     pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG000000000003  747.1942    -0.3507030  0.168246 -2.084470 0.0371175
## ENSG000000000005   0.0000      NA        NA        NA        NA
## ENSG000000000419  520.1342    0.2061078  0.101059  2.039475 0.0414026
## ENSG000000000457  322.6648    0.0245269  0.145145  0.168982 0.8658106
## ENSG000000000460   87.6826    -0.1471420  0.257007 -0.572521 0.5669691
## ...
##           ...       ...       ...       ...       ...
## ENSG00000283115   0.000000      NA        NA        NA        NA
## ENSG00000283116   0.000000      NA        NA        NA        NA
## ENSG00000283119   0.000000      NA        NA        NA        NA
## ENSG00000283120   0.974916    -0.668258   1.69456 -0.394354 0.693319
## ENSG00000283123   0.000000      NA        NA        NA        NA
##           padj
##           <numeric>
## ENSG000000000003   0.163035
## ENSG000000000005   NA
## ENSG000000000419   0.176032
## ENSG000000000457   0.961694
## ENSG000000000460   0.815849
## ...
##           ...
## ENSG00000283115   NA
## ENSG00000283116   NA
## ENSG00000283119   NA
## ENSG00000283120   NA
## ENSG00000283123   NA

```

Let's take a look at the summary of results:

```

summary(res)

##
## out of 25258 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 1563, 6.2%
## LFC < 0 (down)    : 1188, 4.7%
## outliers [1]       : 142, 0.56%
## low counts [2]     : 9971, 39%
## (mean count < 10)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

```

Change the p-value to 0.05 from the 0.1 default:

```

res05 <- results(dds, alpha=0.05)
summary(res05)

##
## out of 25258 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)      : 1236, 4.9%
## LFC < 0 (down)    : 933, 3.7%
## outliers [1]       : 142, 0.56%
## low counts [2]     : 9033, 36%
## (mean count < 6)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

```

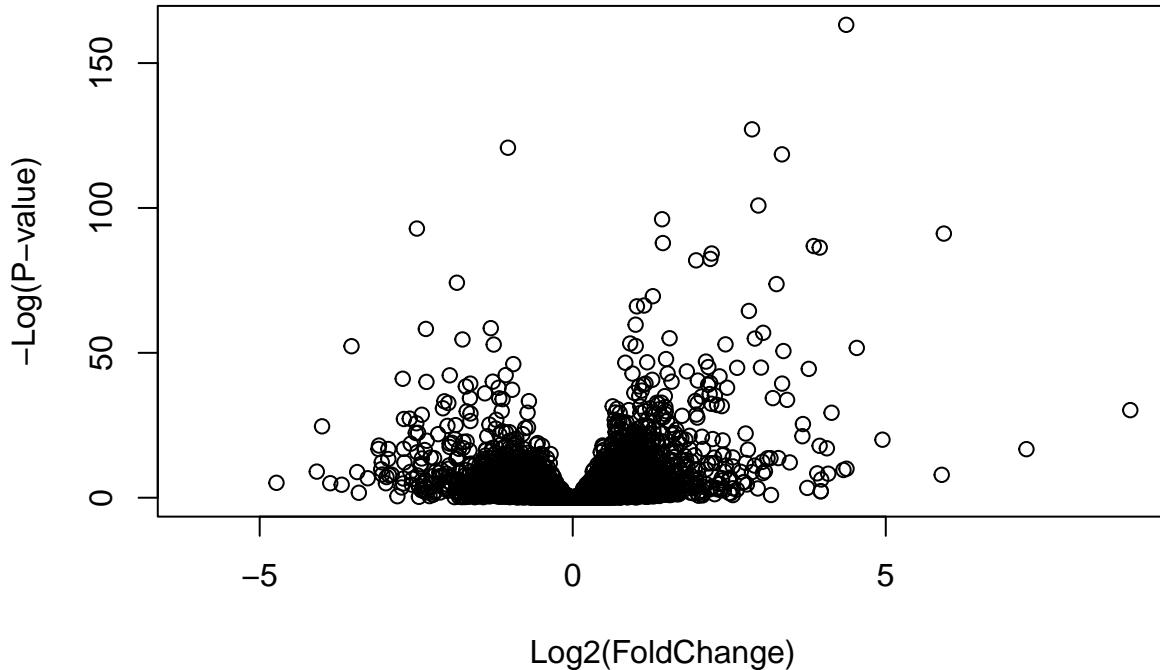
## A volcano plot

This is a very common data viz of this type of data that does not really look like a volcano.

```

plot( res$log2FoldChange, -log(res$padj),
      xlab="Log2(FoldChange)",
      ylab="-Log(P-value)")

```

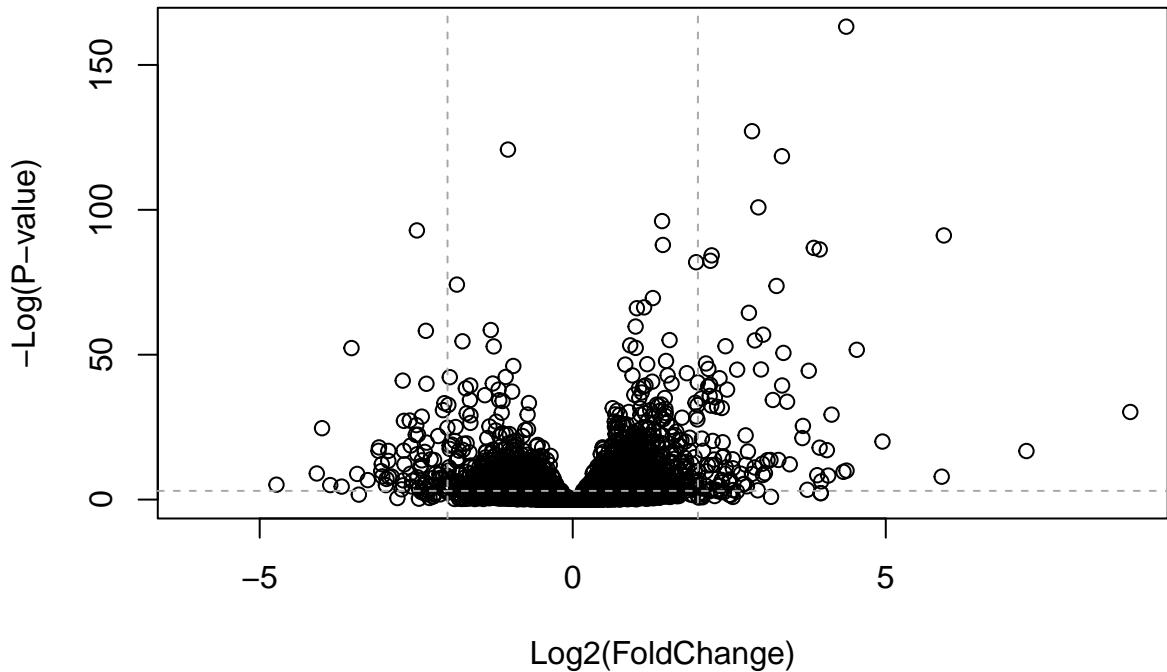


```

plot( res$log2FoldChange, -log(res$padj),
      ylab="-Log(P-value)", xlab="Log2(FoldChange)")

# Add some cut-off lines
abline(v=c(-2,2), col="darkgray", lty=2)
abline(h=-log(0.05), col="darkgray", lty=2)

```



```

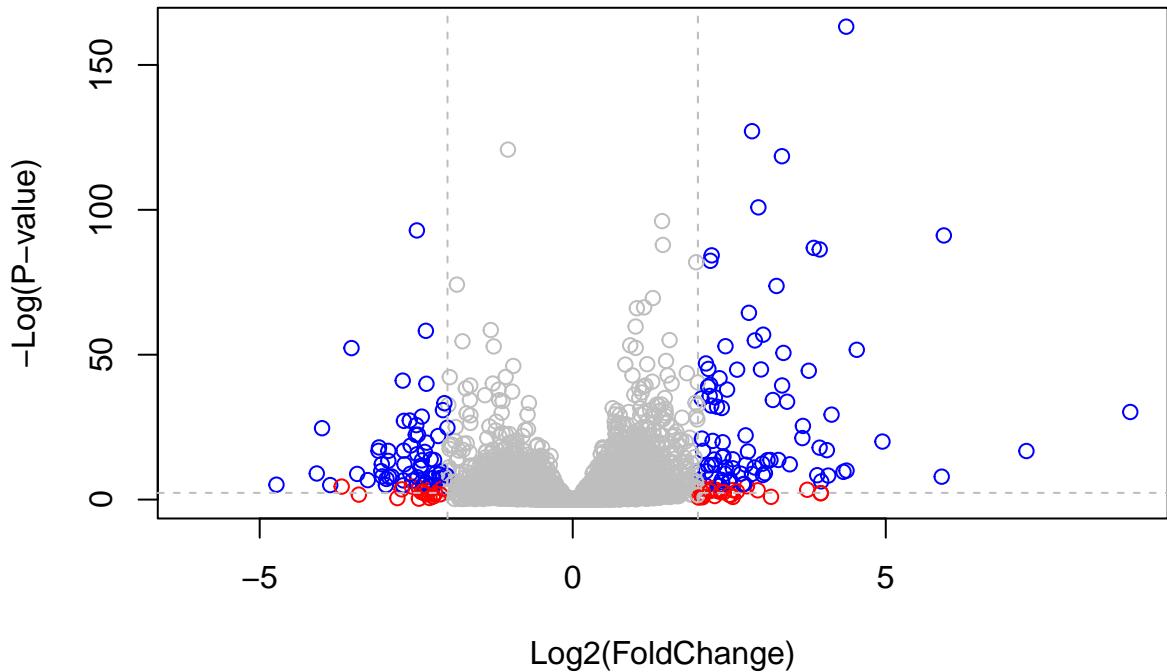
# Setup our custom point color vector
mycols <- rep("gray", nrow(res))
mycols[ abs(res$log2FoldChange) > 2 ] <- "red"

inds <- (res$padj < 0.01) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

# Volcano plot with custom colors
plot( res$log2FoldChange, -log(res$padj),
      col=mycols, ylab="-Log(P-value)", xlab="Log2(FoldChange)" )

# Cut-off lines
abline(v=c(-2,2), col="gray", lty=2)
abline(h=-log(0.1), col="gray", lty=2)

```



Now let's add in some labels:

```
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"
```

```
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
```

```
ord <- order( res$padj )
#View(res[ord,])
head(res[ord,])
```

```
## 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>
## ENSG00000152583 954.771 4.36836 0.2371268 18.4220 8.74490e-76
## ENSG00000179094 743.253 2.86389 0.1755693 16.3120 8.10784e-60
## ENSG00000116584 2277.913 -1.03470 0.0650984 -15.8944 6.92855e-57
## ENSG00000189221 2383.754 3.34154 0.2124058 15.7319 9.14433e-56
## ENSG00000120129 3440.704 2.96521 0.2036951 14.5571 5.26424e-48
## ENSG00000148175 13493.920 1.42717 0.1003890 14.2164 7.25128e-46
## padj symbol
## <numeric> <character>
## ENSG00000152583 1.32441e-71 SPARCL1
## ENSG00000179094 6.13966e-56 PER1
## ENSG00000116584 3.49776e-53 ARHGEF2
## ENSG00000189221 3.46227e-52 MAOA
## ENSG00000120129 1.59454e-44 DUSP1
## ENSG00000148175 1.83034e-42 STOM

```

Store the data:

```
write.csv(res[ord,], "deseq_results.csv")
```

And then do enhanced volcano plot:

```

x <- as.data.frame(res)

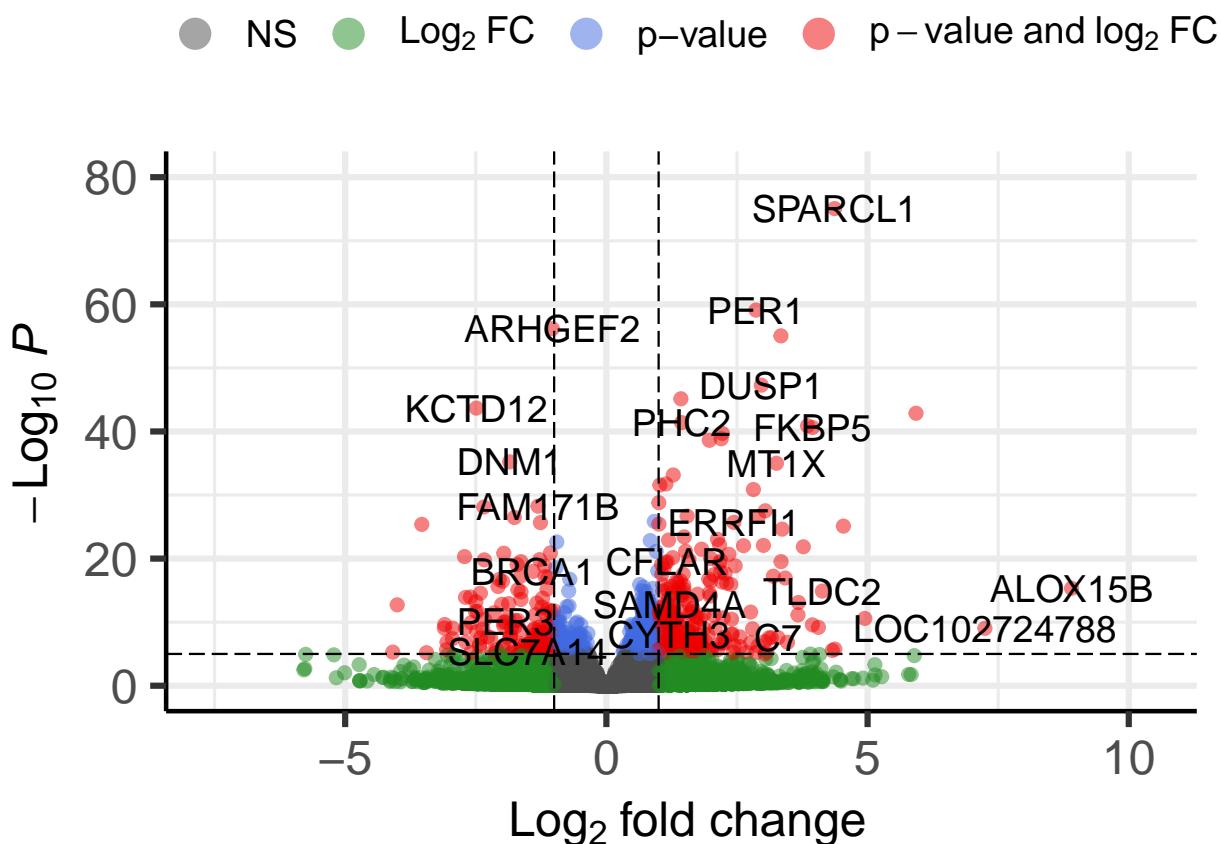
EnhancedVolcano(x,
  lab = x$symbol,
  x = 'log2FoldChange',
  y = 'pvalue')

## Warning: Ignoring unknown parameters: xlim, ylim

```

# Volcano plot

## Enhanced Volcano



```
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)
```

Before we can use KEGG we need to get our gene identifiers in the correct format for KEGG< which is ENTREZ format in this case.

```
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
```

```
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
```

```
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
```

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

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

```
## Info: Working in directory /Users/ShitianLi/OneDrive/GradClass/BGGN 213 Bioinfo/R/BGGN213/class15
```

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

