class12: RNA_Seq Mini Project

Ari_Fon (PID: A15390446)

2/24/2022

Here we will work on a complete differential expression analysis project. We will use DESeq2 for this. First we must load the library

```
library(DESeq2)
library(ggplot2)
library(AnnotationDbi)
library(org.Hs.eg.db)
library(EnhancedVolcano)
```

1. Input the counts and metadata files

```
countData <- read.csv("GSE37704_featurecounts.csv", row.names = 1)
colData <- read.csv("GSE37704_metadata.csv")
```

Inspect these objects

```
colData
```

```
##          id     condition
## 1 SRR493366 control_sirna
## 2 SRR493367 control_sirna
## 3 SRR493368 control_sirna
## 4 SRR493369      hoxa1_kd
## 5 SRR493370      hoxa1_kd
## 6 SRR493371      hoxa1_kd
```

```
head(countData[, -1])
```

```
##                 SRR493366 SRR493367 SRR493368 SRR493369 SRR493370 SRR493371
## ENSG00000186092         0         0         0         0         0         0
## ENSG00000279928         0         0         0         0         0         0
## ENSG00000279457        23        28        29        29        28        46
## ENSG00000278566         0         0         0         0         0         0
## ENSG00000273547         0         0         0         0         0         0
## ENSG00000187634       124       123       205       207       212       258
```

Q. Complete the code below to remove the troublesome first column from countData

```r
countData <- countData[, -1]
head(countData)
```

```
##                 SRR493366 SRR493367 SRR493368 SRR493369 SRR493370 SRR493371
## ENSG00000186092         0         0         0         0         0         0
## ENSG00000279928         0         0         0         0         0         0
## ENSG00000279457        23        28        29        29        28        46
## ENSG00000278566         0         0         0         0         0         0
## ENSG00000273547         0         0         0         0         0         0
## ENSG00000187634       124       123       205       207       212       258
```

```r
colData$id == colnames(countData)
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE
```

> Q. Check on corespodence of colData and countData

```r
all(colData$id == colnames(countData))
```

```
## [1] TRUE
```

> Q. Complete the code below to filter countData to exclude genes (i.e. rows) where we have 0 read count across all samples (i.e. columns).

```r
head(countData)
```

```
##                 SRR493366 SRR493367 SRR493368 SRR493369 SRR493370 SRR493371
## ENSG00000186092         0         0         0         0         0         0
## ENSG00000279928         0         0         0         0         0         0
## ENSG00000279457        23        28        29        29        28        46
## ENSG00000278566         0         0         0         0         0         0
## ENSG00000273547         0         0         0         0         0         0
## ENSG00000187634       124       123       205       207       212       258
```

```r
counts <- countData[rowSums(countData) != 0, ]
head(counts)
```

```
##                 SRR493366 SRR493367 SRR493368 SRR493369 SRR493370 SRR493371
## ENSG00000279457        23        28        29        29        28        46
## ENSG00000187634       124       123       205       207       212       258
## ENSG00000188976      1637      1831      2383      1226      1326      1504
## ENSG00000187961       120       153       180       236       255       357
## ENSG00000187583        24        48        65        44        48        64
## ENSG00000187642         4         9        16        14        16        16
```

```r
nrow(counts)
```

```
## [1] 15975
```

Running DESeq2 Analysis

```
dds <- DESeqDataSetFromMatrix(countData=counts,
                              colData=colData,
                              design=~condition)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

Now I can run my differential expression with `DESeq()`

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

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

Now get my results out of this dds object

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

```
## log2 fold change (MLE): condition hoxa1 kd vs control sirna
## Wald test p-value: condition hoxa1 kd vs control sirna
## DataFrame with 15975 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE       stat      pvalue
##                 <numeric>      <numeric> <numeric>  <numeric>   <numeric>
## ENSG00000279457   29.9136      0.1792571 0.3248216   0.551863 5.81042e-01
## ENSG00000187634  183.2296      0.4264571 0.1402658   3.040350 2.36304e-03
## ENSG00000188976 1651.1881     -0.6927205 0.0548465 -12.630158 1.43990e-36
## ENSG00000187961  209.6379      0.7297556 0.1318599   5.534326 3.12428e-08
## ENSG00000187583   47.2551      0.0405765 0.2718928   0.149237 8.81366e-01
## ...                   ...            ...       ...        ...         ...
## ENSG00000273748  35.30265       0.674387  0.303666   2.220817 2.63633e-02
## ENSG00000278817   2.42302      -0.388988  1.130394  -0.344117 7.30758e-01
## ENSG00000278384   1.10180       0.332991  1.660261   0.200565 8.41039e-01
## ENSG00000276345  73.64496      -0.356181  0.207716  -1.714752 8.63908e-02
## ENSG00000271254 181.59590      -0.609667  0.141320  -4.314071 1.60276e-05
##                        padj
##                   <numeric>
## ENSG00000279457 6.86555e-01
## ENSG00000187634 5.15718e-03
## ENSG00000188976 1.76549e-35
## ENSG00000187961 1.13413e-07
## ENSG00000187583 9.19031e-01
## ...                     ...
## ENSG00000273748 4.79091e-02
## ENSG00000278817 8.09772e-01
## ENSG00000278384 8.92654e-01
## ENSG00000276345 1.39762e-01
## ENSG00000271254 4.53648e-05
```

Add annotation

> Q. Use the mapIDs() function multiple times to add SYMBOL, ENTREZID and GENENAME annotation to our results by completing the code below.

```
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),
                    keytype="ENSEMBL",
                    column="SYMBOL",
                    multiVals="first")
```

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

```
res$entrez <- mapIds(org.Hs.eg.db,
                    keys=row.names(res),
                    keytype="ENSEMBL",
                    column="ENTREZID",
                    multiVals="first")
```

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

```
res$name <- mapIds(org.Hs.eg.db,
                    keys=row.names(res),
                    keytype="ENSEMBL",
                    column="GENENAME",
                    multiVals="first")
```

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

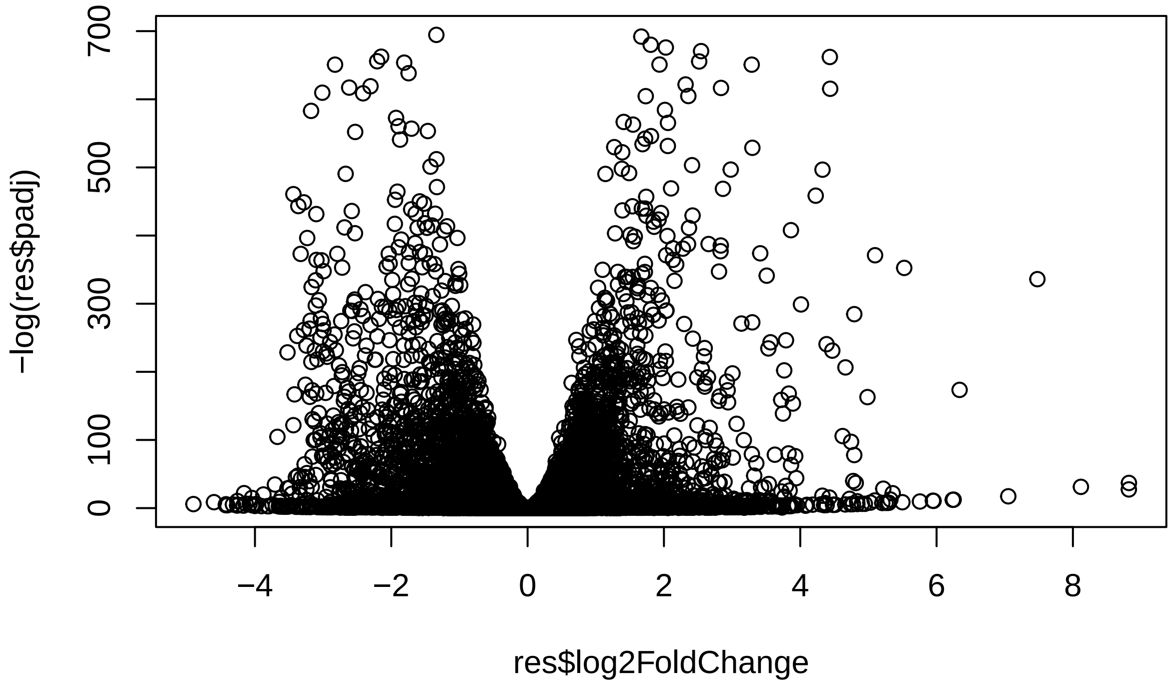
```
head(res)
```

```
## log2 fold change (MLE): condition hoxa1 kd vs control sirna
## Wald test p-value: condition hoxa1 kd vs control sirna
## DataFrame with 6 rows and 9 columns
##                  baseMean log2FoldChange     lfcSE       stat      pvalue
##                 <numeric>      <numeric> <numeric>  <numeric>   <numeric>
## ENSG00000279457   29.9136      0.1792571 0.3248216   0.551863 5.81042e-01
## ENSG00000187634  183.2296      0.4264571 0.1402658   3.040350 2.36304e-03
## ENSG00000188976 1651.1881     -0.6927205 0.0548465 -12.630158 1.43990e-36
## ENSG00000187961  209.6379      0.7297556 0.1318599   5.534326 3.12428e-08
## ENSG00000187583   47.2551      0.0405765 0.2718928   0.149237 8.81366e-01
## ENSG00000187642   11.9798      0.5428105 0.5215598   1.040744 2.97994e-01
##                        padj      symbol      entrez                   name
##                   <numeric> <character> <character>            <character>
## ENSG00000279457 6.86555e-01      WASH9P   102723897 WAS protein family h..
## ENSG00000187634 5.15718e-03      SAMD11      148398 sterile alpha motif ..
## ENSG00000188976 1.76549e-35       NOC2L       26155 NOC2 like nucleolar ..
## ENSG00000187961 1.13413e-07      KLHL17      339451 kelch like family me..
## ENSG00000187583 9.19031e-01     PLEKHN1       84069 pleckstrin homology ..
## ENSG00000187642 4.03379e-01       PERM1       84808 PPARGC1 and ESRR ind..
```

Volcano Plot

Common summary figure that gives a nice overview of our results

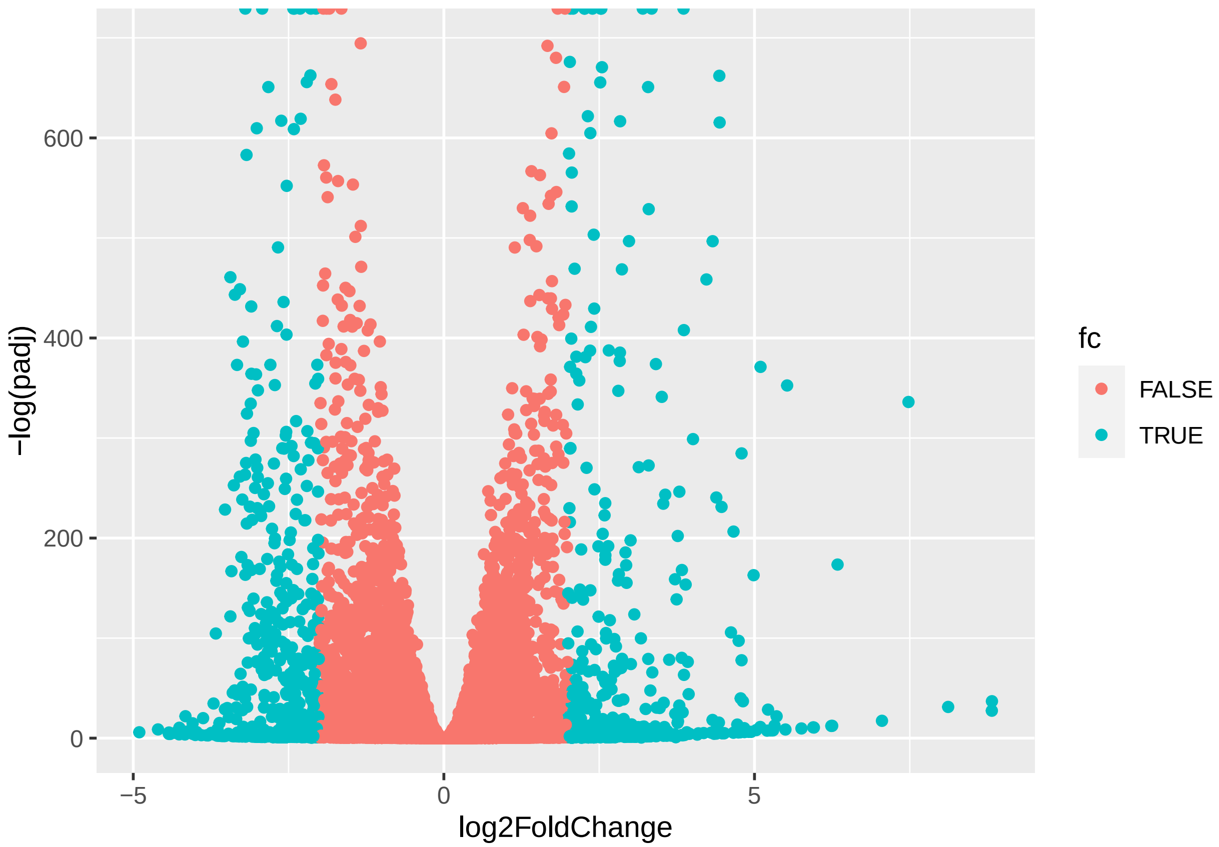
```
plot(res$log2FoldChange, -log(res$padj))
```



Try ggplot for this

```
tmp <- as.data.frame(res)
tmp$fc <- abs(res$log2FoldChange) > 2
ggplot(tmp) +
  aes(log2FoldChange, -log(padj), col=fc) +
  geom_point()
```

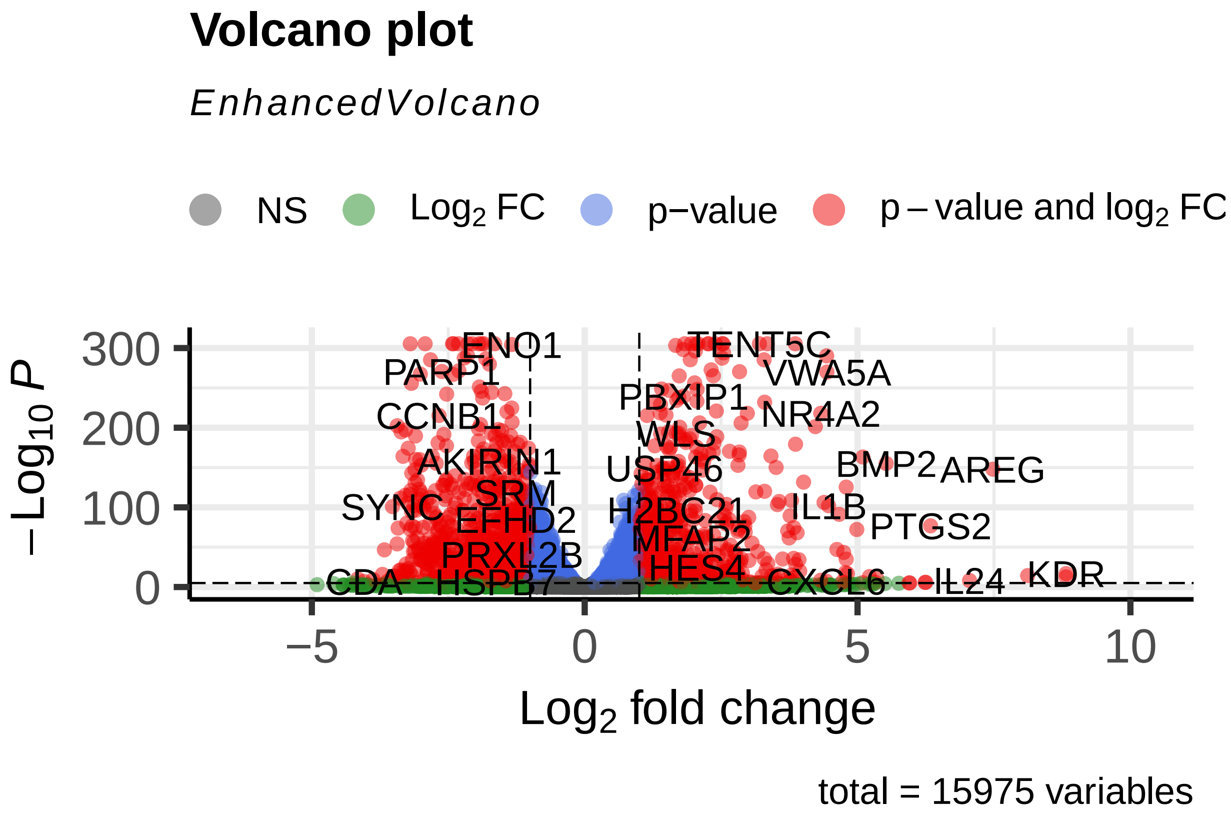
```
## Warning: Removed 1237 rows containing missing values (geom_point).
```



Try EnhancedVolcano package from Bioconductor

```
tmp <- as.data.frame(res)
EnhancedVolcano(tmp,
    lab = tmp$symbol,
    x = 'log2FoldChange',
    y = 'pvalue')
```

```
## Warning: One or more p-values is 0. Converting to 10^-1 * current lowest non-
## zero p-value...
```



#Pathway analysis and gene set enrichment

Here we try to bring back the biology and help with the interpretation of our results. We try to answer the question : which pathways and functions feature heavily in our differentially expressed genes?

Recall that we need a "vector of importance" as input for GAGE that has ENTREZ ids as the names attributes

```
foldchange <- res$log2FoldChange
names(foldchange) <- res$entrez
```

```
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)
data(sigmet.idx.hs)
```

```
keggres = gage(foldchange, gsets=kegg.sets.hs)
```

Look at the first 2 down-regulated pathways

```
# Look at the first few down (less) pathways
head(keggres$less, 2)
```

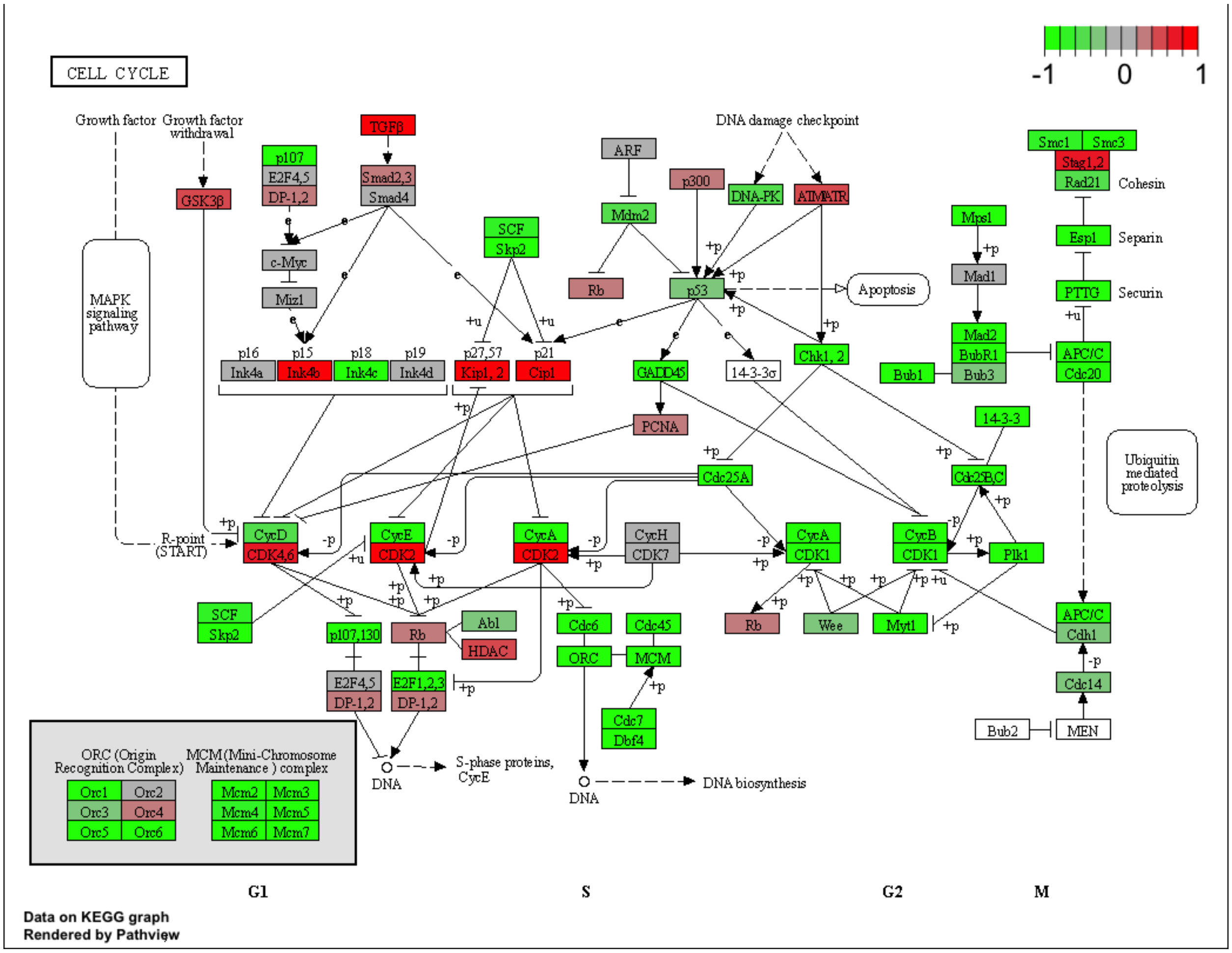
```
##                              p.geomean stat.mean        p.val       q.val
## hsa04110 Cell cycle      8.995727e-06 -4.378644 8.995727e-06 0.001889103
## hsa03030 DNA replication 9.424076e-05 -3.951803 9.424076e-05 0.009841047
##                          set.size         exp1
## hsa04110 Cell cycle           121 8.995727e-06
## hsa03030 DNA replication       36 9.424076e-05
```

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

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

```
## Info: Working in directory /Users/Ari_Fon/Desktop/BIMM143 /class12
```

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



Gene Ontology Analysis

We can use a different gene set data base (we used KEGG above) to provide different (but hopefully complementary) information. We will try GO here with a focus on Biological Pathways (BP) component

Look at the GO sets

```
data(go.sets.hs)
data(go.subs.hs)

# Focus on Biological Process subset of GO
gobpsets = go.sets.hs[go.subs.hs$BP]

gobpres = gage(foldchange, gsets=gobpsets, same.dir=TRUE)

head(gobpres$less)
```

```
##                                             p.geomean stat.mean        p.val
## GO:0048285 organelle fission             1.536227e-15 -8.063910 1.536227e-15
## GO:0000280 nuclear division              4.286961e-15 -7.939217 4.286961e-15
## GO:0007067 mitosis                       4.286961e-15 -7.939217 4.286961e-15
## GO:0000087 M phase of mitotic cell cycle 1.169934e-14 -7.797496 1.169934e-14
## GO:0007059 chromosome segregation        2.028624e-11 -6.878340 2.028624e-11
## GO:0000236 mitotic prometaphase          1.729553e-10 -6.695966 1.729553e-10
##                                                 q.val set.size         exp1
## GO:0048285 organelle fission             5.841698e-12      376 1.536227e-15
## GO:0000280 nuclear division              5.841698e-12      352 4.286961e-15
## GO:0007067 mitosis                       5.841698e-12      352 4.286961e-15
## GO:0000087 M phase of mitotic cell cycle 1.195672e-11      362 1.169934e-14
## GO:0007059 chromosome segregation        1.658603e-08      142 2.028624e-11
## GO:0000236 mitotic prometaphase          1.178402e-07       84 1.729553e-10
```

Reactome

We can use Reactome either as an R package (just like above) or we can use the website. The wbsite needs a file of "gene important" just like gage above.

Reactome is database consisting of biological molecules and their relation to pathways and processes. Reactome, such as many other tools, has an online software available (https://reactome.org/) and R package available (https://bioconductor.org/packages/release/bioc/html/ReactomePA.html).

```r
sig_genes <- res[res$padj <= 0.05 & !is.na(res$padj), "symbol"]

write.table(sig_genes, file="significant_genes.txt", row.names=FALSE, col.names=FALSE, quote=FALSE)
```

Save my results

```r
write.csv(res, file ="deseq_results.csv")
```