Class 14: RNASeq mini project

Serena Quezada

Table of contents

Background

Here we work through a complete RNASeq analysis project

Data Import

Reading the `counts` and `metadata` CSV files

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

Check on data structure

```
head(rawcounts)
```

```
                length SRR493366 SRR493367 SRR493368 SRR493369 SRR493370
ENSG00000186092    918         0         0         0         0         0
ENSG00000279928    718         0         0         0         0         0
ENSG00000279457   1982        23        28        29        29        28
ENSG00000278566    939         0         0         0         0         0
ENSG00000273547    939         0         0         0         0         0
ENSG00000187634   3214       124       123       205       207       212
                SRR493371
ENSG00000186092         0
ENSG00000279928         0
ENSG00000279457        46
ENSG00000278566         0
ENSG00000273547         0
ENSG00000187634       258
```

```
metadata
```

```
         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
```

Some book-keeping is required as there looks to be a mis-match between metadata rows and counts columns.

```
ncol(rawcounts)
```

```
[1] 7
```

```
nrow(metadata)
```

```
[1] 6
```

Looks like we need to get rid of the first "length" column of our `counts` object.

```
counts <- rawcounts[, -1]
```

```
cleancounts <- counts[rowSums(counts) > 0, ]
```

```
# Keep only columns in cleancounts that match metadata$id
cleancounts <- cleancounts[, metadata$id]

# Optionally, reorder metadata to match the column order of cleancounts
metadata <- metadata[match(colnames(cleancounts), metadata$id), ]
```

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

```
[1] TRUE
```

```
metadata$id
```

```
[1] "SRR493366" "SRR493367" "SRR493368" "SRR493369" "SRR493370" "SRR493371"
```

Remove zero count genes

There are lots of genes with zero counts. We can remove these from further analysis.

```
head(cleancounts)
```

	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

```
to.keep.inds <- rowSums(cleancounts) > 0
nonzero_counts <- cleancounts[to.keep.inds, ]
```

DESeq analysis

Load the package

```
library(DESeq2)
```

Set up DESeq object

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

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

Run DEseq

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

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

Get results

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

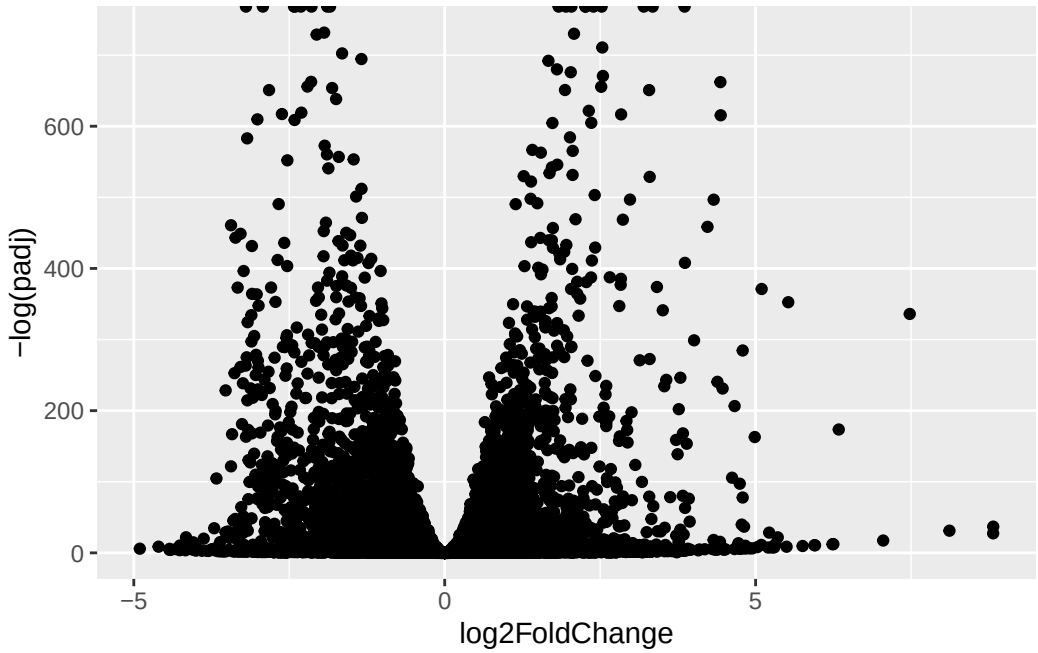
```
log2 fold change (MLE): condition hoxa1 kd vs control sirna
Wald test p-value: condition hoxa1 kd vs control sirna
DataFrame with 6 rows and 6 columns
                 baseMean log2FoldChange     lfcSE       stat      pvalue
                <numeric>      <numeric> <numeric>  <numeric>   <numeric>
ENSG00000279457   29.9136      0.1792571 0.3248215   0.551863 5.81042e-01
ENSG00000187634  183.2296      0.4264571 0.1402658   3.040350 2.36304e-03
ENSG00000188976 1651.1881     -0.6927205 0.0548465 -12.630156 1.43993e-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
                 <numeric>
ENSG00000279457 6.86555e-01
ENSG00000187634 5.15718e-03
ENSG00000188976 1.76553e-35
ENSG00000187961 1.13413e-07
ENSG00000187583 9.19031e-01
ENSG00000187642 4.03379e-01
```

Data Visualization

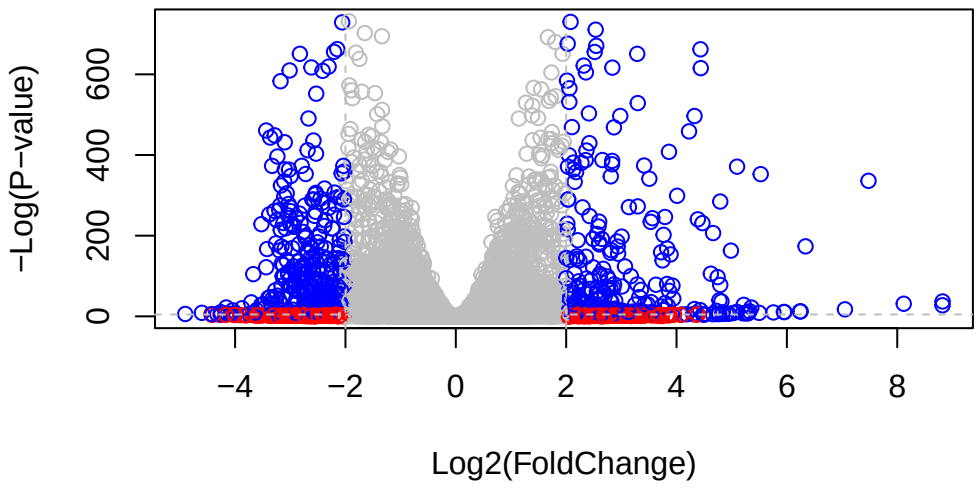
Volcano plot

```
library(ggplot2)

ggplot(res) +
  aes(log2FoldChange, -log(padj)) +
  geom_point()
```

```
Warning: Removed 1237 rows containing missing values or values outside the scale range
(`geom_point()`).
```




```
mycols <- rep("gray", nrow(res))
mycols[ abs(res$log2FoldChange) > 2 ] <- "red"
inds <- (res$padj < 0.01) & (abs(res$log2FoldChange) > 2)
mycols[inds] <- "blue"

plot(res$log2FoldChange, -log(res$padj), col=mycols,
     xlab="Log2(FoldChange)", ylab="-Log(P-value)")
abline(v=c(-2,2), col="gray", lty=2)
abline(h=-log(0.01), col="gray", lty=2)
```



Add threshold lines for fold-change and P-value and color our subset of genes that makes these threshol cut-offs in the plot

Add Annotation

Add gene symbol and entrez

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

```
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 (x= org.Hs.eg.db,
        keys = row.names(res),
        keytype = "ENSEMBL",
        column = "SYMBOL")
```

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

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

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

Pathway Analysis

Run gage analysis

```
library(gage)
library(gageData)
library(pathview)

data(kegg.sets.hs)
data(sigmet.idx.hs)

# Focus on signaling and metabolic pathways only
kegg.sets.hs = kegg.sets.hs[sigmet.idx.hs]

# Examine the first 3 pathways
head(kegg.sets.hs, 3)
```

```
$`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"

$`hsa00230 Purine metabolism`
  [1] "100"    "10201"  "10606"  "10621"  "10622"  "10623"  "107"    "10714"
  [9] "108"    "10846"  "109"    "111"    "11128"  "11164"  "112"    "113"
 [17] "114"    "115"    "122481" "122622" "124583" "132"    "158"    "159"
 [25] "1633"   "171568" "1716"   "196883" "203"    "204"    "205"    "221823"
 [33] "2272"   "22978"  "23649"  "246721" "25885"  "2618"   "26289"  "270"
 [41] "271"    "27115"  "272"    "2766"   "2977"   "2982"   "2983"   "2984"
 [49] "2986"   "2987"   "29922"  "3000"   "30833"  "30834"  "318"    "3251"
 [57] "353"    "3614"   "3615"   "3704"   "377841" "471"    "4830"   "4831"
 [65] "4832"   "4833"   "4860"   "4881"   "4882"   "4907"   "50484"  "50940"
 [73] "51082"  "51251"  "51292"  "5136"   "5137"   "5138"   "5139"   "5140"
 [81] "5141"   "5142"   "5143"   "5144"   "5145"   "5146"   "5147"   "5148"
 [89] "5149"   "5150"   "5151"   "5152"   "5153"   "5158"   "5167"   "5169"
 [97] "51728"  "5198"   "5236"   "5313"   "5315"   "53343"  "54107"  "5422"
[105] "5424"   "5425"   "5426"   "5427"   "5430"   "5431"   "5432"   "5433"
[113] "5434"   "5435"   "5436"   "5437"   "5438"   "5439"   "5440"   "5441"
[121] "5471"   "548644" "55276"  "5557"   "5558"   "55703"  "55811"  "55821"
[129] "5631"   "5634"   "56655"  "56953"  "56985"  "57804"  "58497"  "6240"
[137] "6241"   "64425"  "646625" "654364" "661"    "7498"   "8382"   "84172"
[145] "84265"  "84284"  "84618"  "8622"   "8654"   "87178"  "8833"   "9060"
[153] "9061"   "93034"  "953"    "9533"   "954"    "955"    "956"    "957"
[161] "9583"   "9615"
```

We need a named vector of fold-change values as input for gage

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

```
       <NA>      148398       26155      339451       84069       84808
 0.17925708  0.42645712 -0.69272046  0.72975561  0.04057653  0.54281049
```

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

```
head(keggres$less)
```

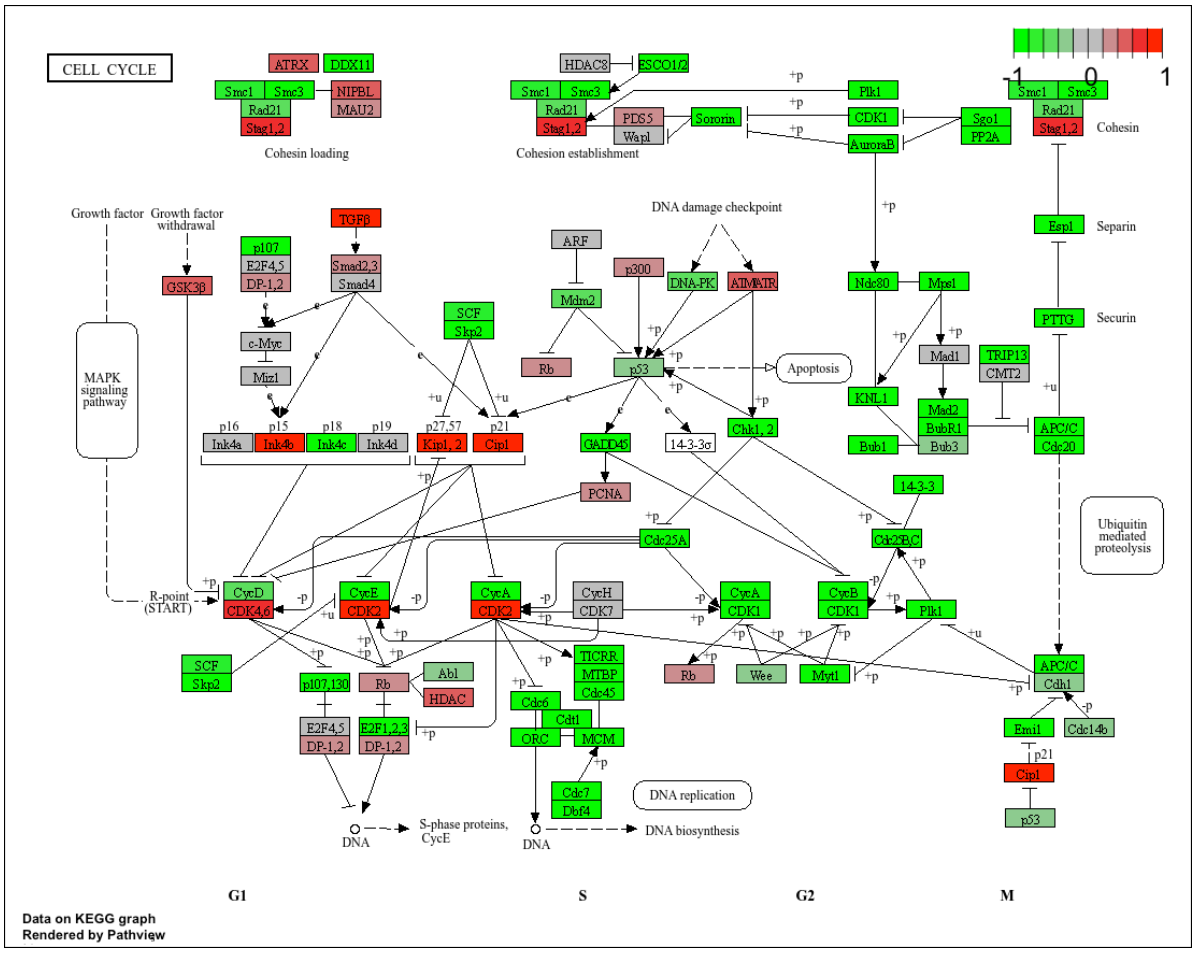
```
                                                    p.geomean stat.mean
hsa04110 Cell cycle                                8.995727e-06 -4.378644
hsa03030 DNA replication                           9.424076e-05 -3.951803
hsa05130 Pathogenic Escherichia coli infection 1.405864e-04 -3.765330
hsa03013 RNA transport                             1.246882e-03 -3.059466
hsa03440 Homologous recombination                  3.066756e-03 -2.852899
hsa04114 Oocyte meiosis                            3.784520e-03 -2.698128
                                                           p.val       q.val
hsa04110 Cell cycle                                8.995727e-06 0.001889103
hsa03030 DNA replication                           9.424076e-05 0.009841047
hsa05130 Pathogenic Escherichia coli infection 1.405864e-04 0.009841047
hsa03013 RNA transport                             1.246882e-03 0.065461280
hsa03440 Homologous recombination                  3.066756e-03 0.128803765
hsa04114 Oocyte meiosis                            3.784520e-03 0.132458190
                                                   set.size         exp1
hsa04110 Cell cycle                                     121 8.995727e-06
hsa03030 DNA replication                                 36 9.424076e-05
hsa05130 Pathogenic Escherichia coli infection           53 1.405864e-04
hsa03013 RNA transport                                  144 1.246882e-03
hsa03440 Homologous recombination                        28 3.066756e-03
hsa04114 Oocyte meiosis                                 102 3.784520e-03
```

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

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

```
Info: Working in directory /Users/serenaquezada/Library/Mobile Documents/com~apple~CloudDocs/
```

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

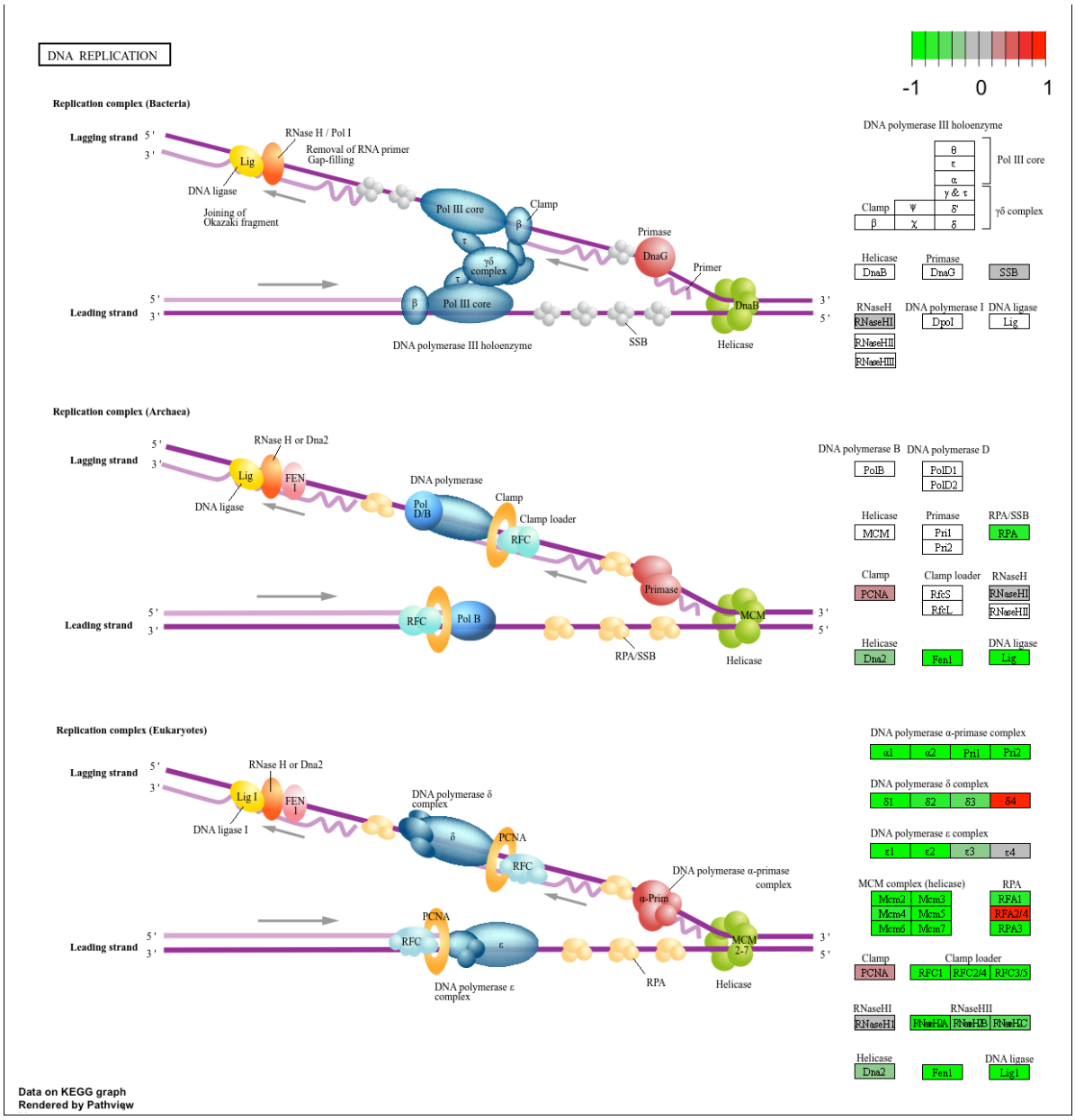



```
pathview(pathway.id = "hsa03030", gene.data = foldchanges)
```

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

```
Info: Working in directory /Users/serenaquezada/Library/Mobile Documents/com~apple~CloudDocs/
```

```
Info: Writing image file hsa03030.pathview.png
```



GO terms

Same analysis but using GO genesets rathen than KEGG

```
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(foldchanges, gsets=gobpsets, same.dir=TRUE)
```

```
head(gobpres$less, 4)
```

```
                                      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
                                            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
```

Reactome

Lots of people like the reactome web interface. You can also run this as an R function but lets look at the website first < https://reactome.org/ >

The website wants a text file with one gene symbol per line of the genes you want to map to pathways

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

```
ENSG00000187634 ENSG00000188976 ENSG00000187961 ENSG00000188290 ENSG00000187608
       "SAMD11"         "NOC2L"        "KLHL17"          "HES4"         "ISG15"
ENSG00000188157
         "AGRN"
```

and write out a file:

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

Save our results

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