# Class 14: RNASeq mini project

Mai Tamura (PID: A18594079)

## Table of contents



### Background

Here we work through a complete RNASeq analysis project. The input data comes from a knock-down experiment of a HOX gene

### Data Import

Reading the `counts` and `metadata` CSV files

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

Check on data structure

```
head(counts)
```

```
                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
```

```
head(metadata)
```

```
              condition
SRR493366 control_sirna
SRR493367 control_sirna
SRR493368 control_sirna
SRR493369      hoxa1_kd
SRR493370      hoxa1_kd
SRR493371      hoxa1_kd
```

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

```
ncol(counts)
```

```
[1] 7
```

```
nrow(metadata)
```

```
[1] 6
```

Looks like we need to get rid of the first "lengths" column of our `counts`

```
cleancounts <- counts[,-1]
head(cleancounts)
```

```
                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
```

```
colnames(cleancounts)
```

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

## Remove zero count genes

There are lots of genes with zero counts so let's get rid of them

```
head(cleancounts)
```

```
                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
```

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

## DESeq analysis

Load the package

```
library(DESeq2)
```

```
Warning: package 'matrixStats' was built under R version 4.5.2
```

Setup 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
```

```
head(dds)
```

```
class: DESeqDataSet
dim: 6 6
metadata(1): version
assays(1): counts
rownames(6): ENSG00000279457 ENSG00000187634 ... ENSG00000187583
  ENSG00000187642
rowData names(0):
colnames(6): SRR493366 SRR493367 ... SRR493370 SRR493371
colData names(1): condition
```

Run DESeq

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

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

```
head(dds)
```

```
class: DESeqDataSet
dim: 6 6
metadata(1): version
assays(4): counts mu H cooks
rownames(6): ENSG00000279457 ENSG00000187634 ... ENSG00000187583
  ENSG00000187642
rowData names(22): baseMean baseVar ... deviance maxCooks
colnames(6): SRR493366 SRR493367 ... SRR493370 SRR493371
colData names(2): condition sizeFactor
```

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 0.3248216   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
                  <numeric>
ENSG00000279457 6.86555e-01
ENSG00000187634 5.15718e-03
ENSG00000188976 1.76549e-35
ENSG00000187961 1.13413e-07
ENSG00000187583 9.19031e-01
ENSG00000187642 4.03379e-01
```

Get a sense of how many genes are up or down-regulated at the default 0.1 p-value cutoff

```
summary(res)
```

```
out of 15975 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 4349, 27%
LFC < 0 (down)     : 4396, 28%
outliers [1]       : 0, 0%
low counts [2]     : 1237, 7.7%
(mean count < 0)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

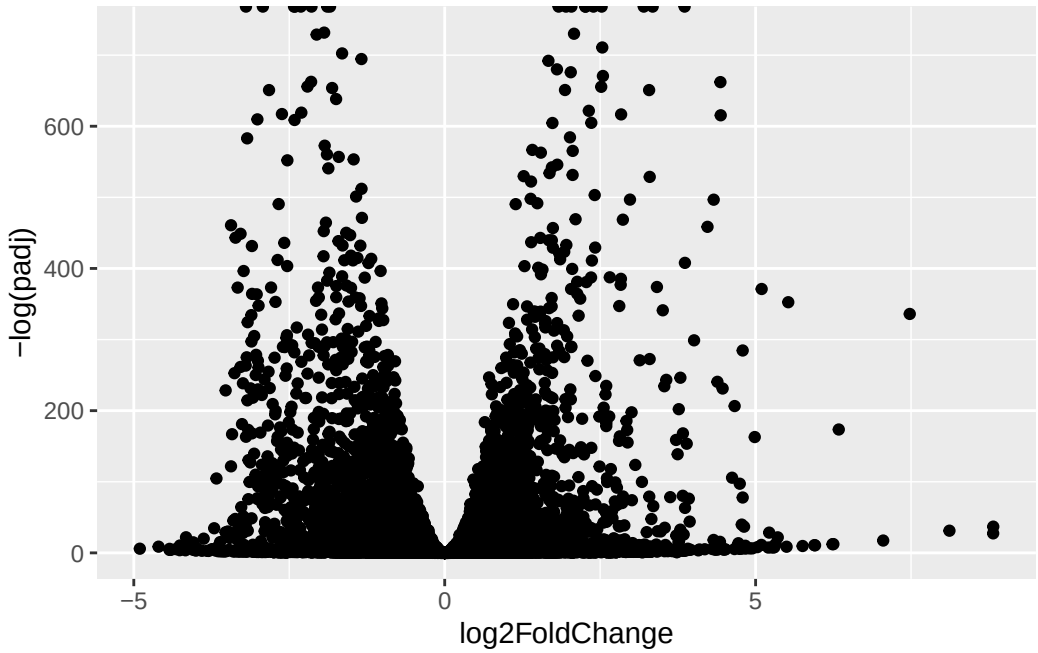
## Data Visualization

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()`).
```

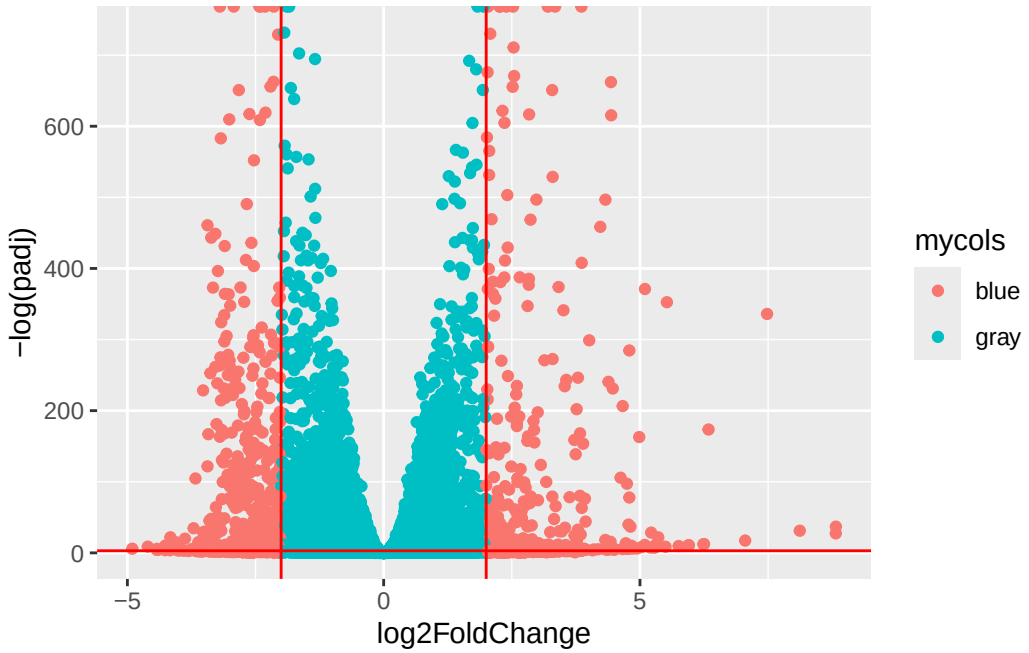



Add threshold lines for fold-change and P-value and color our subset of genes that make these threshold cut-offs in the plot

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

ggplot(res) +
  aes(log2FoldChange, -log(padj), col = mycols) +
  geom_point() +
  geom_hline(yintercept = -log(0.05), col = "red") +
  geom_vline(xintercept = c(2, -2), col = "red")
```

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




## Add Annotation

Add gene symbols and entrez ids

```
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"
```

Add “SYMBOL”, “ENTREZID” and “GENENAME” annotation to our results

```
res$symbol <- mapIds(keys = row.names(res), # our current IDs
                     keytype = "ENSEMBL",   # the format of our IDs
                     x = org.Hs.eg.db,      # where to get the mappings from
                     column = "SYMBOL")     # the format/DB to map to
```

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

```
res$entrez <- mapIds(keys = row.names(res), # our current IDs
                     keytype = "ENSEMBL",   # the format of our IDs
                     x = org.Hs.eg.db,      # where to get the mappings from
                     column = "ENTREZID")   # the format/DB to map to
```

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

```
res$genename <- mapIds(keys = row.names(res), # our current IDs
                       keytype = "ENSEMBL",   # the format of our IDs
                       x = org.Hs.eg.db,      # where to get the mappings from
                       column = "GENENAME")   # the format/DB to map to
```

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

```
head(res, 5)
```

```
log2 fold change (MLE): condition hoxa1 kd vs control sirna
Wald test p-value: condition hoxa1 kd vs control sirna
DataFrame with 5 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
                      padj      symbol      entrez              genename
                 <numeric> <character> <character>           <character>
ENSG00000279457 6.86555e-01          NA          NA                    NA
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 ..
```

Let's reorder these results by adjusted p-value and save them to a CSV file in your current project directory

```
res_reorder <- res[order(res$pvalue),]
write.csv(res_reorder, file="deseq_results.csv")
```

## Pathway Anlysis

### KEGG pathways

Run gage analysis with KEGG

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

We need a names 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, 5)
```

```
                                           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
                                                 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 0.065461279
hsa03440 Homologous recombination                 3.066756e-03 0.128803765
                                                  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
```

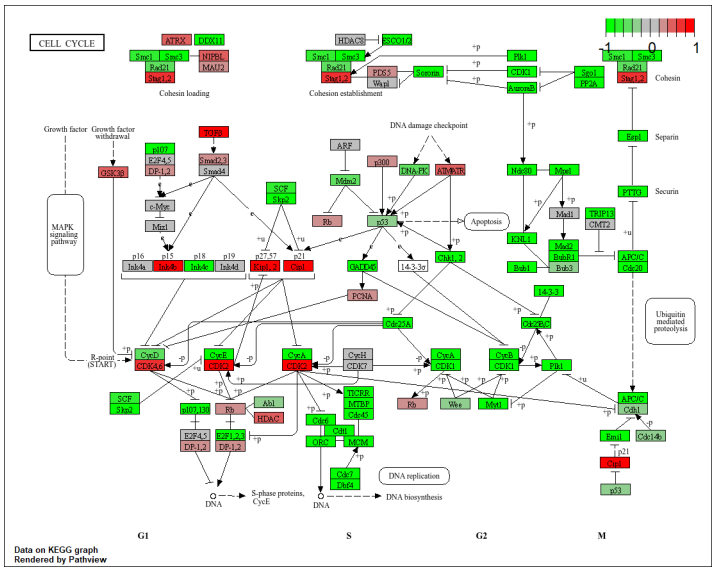
```
pathview(pathway.id = "hsa04110", gene.data = foldchanges)
```

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

```
Info: Working in directory C:/Users/maima/OneDrive/   /School/UCSD/Class/BIMM 143 FA'25/clas
```

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

Add this pathway figure to our lab report



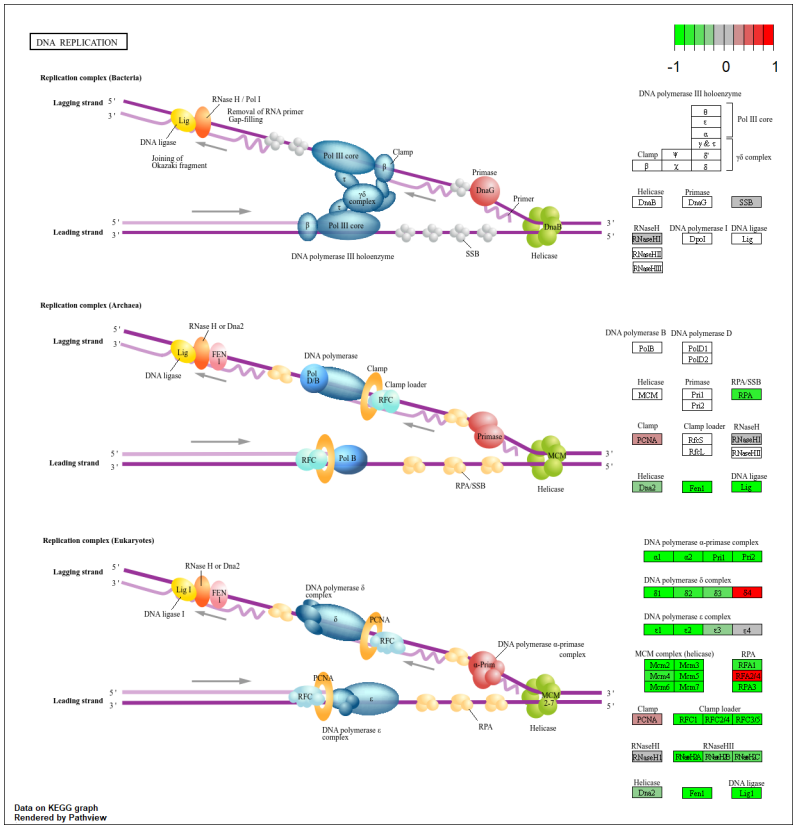

Do the same procedure as above to plot the pathview figures for the top 5 down-reguled pathways

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

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

```
Info: Working in directory C:/Users/maima/OneDrive/   /School/UCSD/Class/BIMM 143 FA'25/clas
```

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

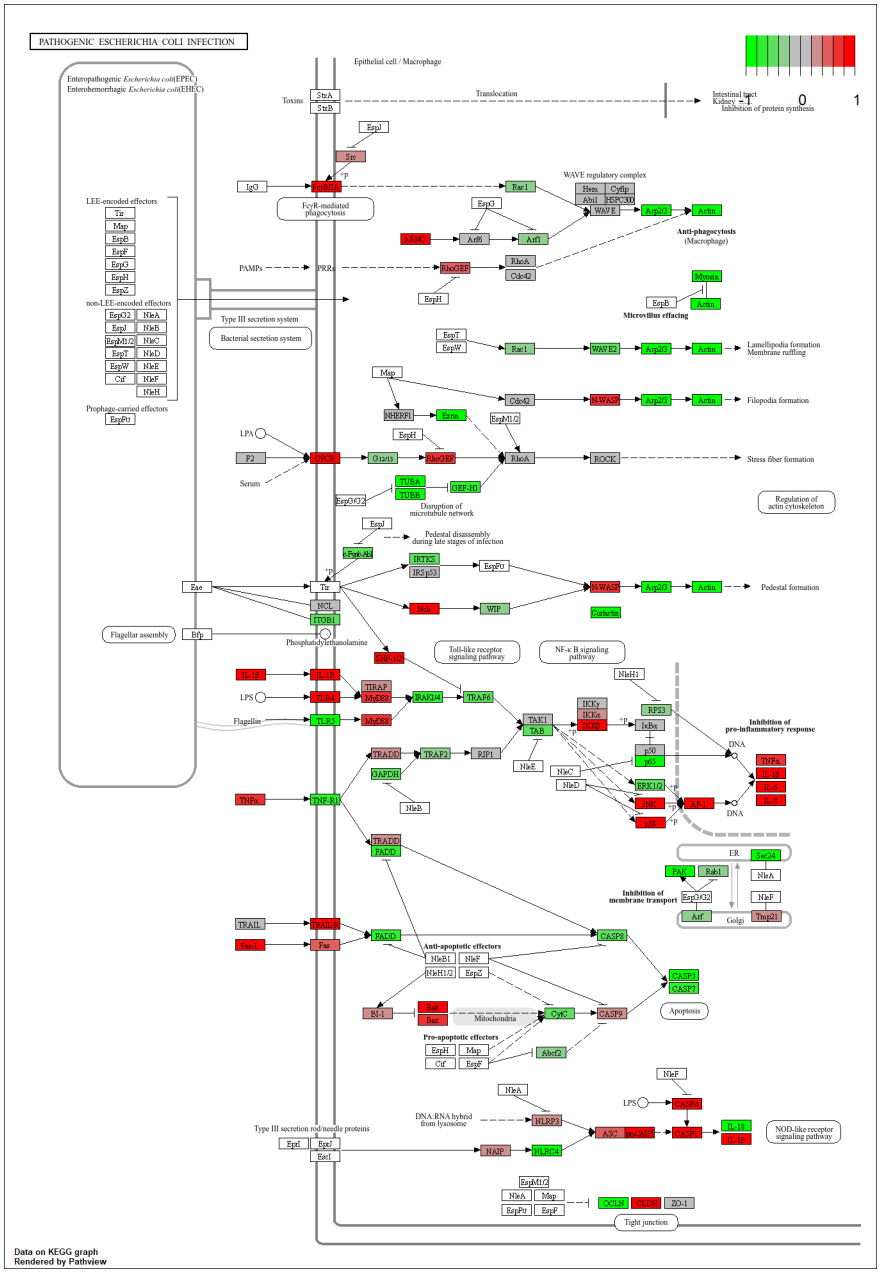



```
pathview(pathway.id = "hsa05130", gene.data = foldchanges)
```

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

```
Info: Working in directory C:/Users/maima/OneDrive/   /School/UCSD/Class/BIMM 143 FA'25/clas
```

```
Info: Writing image file hsa05130.pathview.png
```

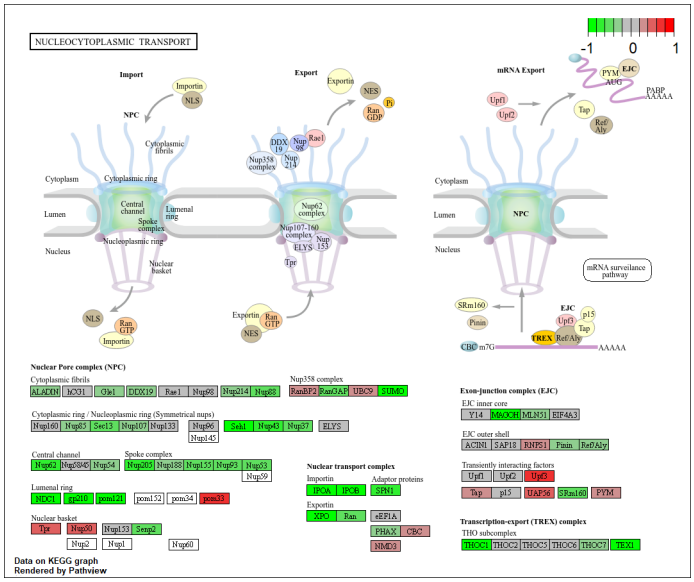



```
pathview(pathway.id = "hsa03013", gene.data = foldchanges)
```

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

```
Info: Working in directory C:/Users/maima/OneDrive/   /School/UCSD/Class/BIMM 143 FA'25/clas
```

```
Info: Writing image file hsa03013.pathview.png
```

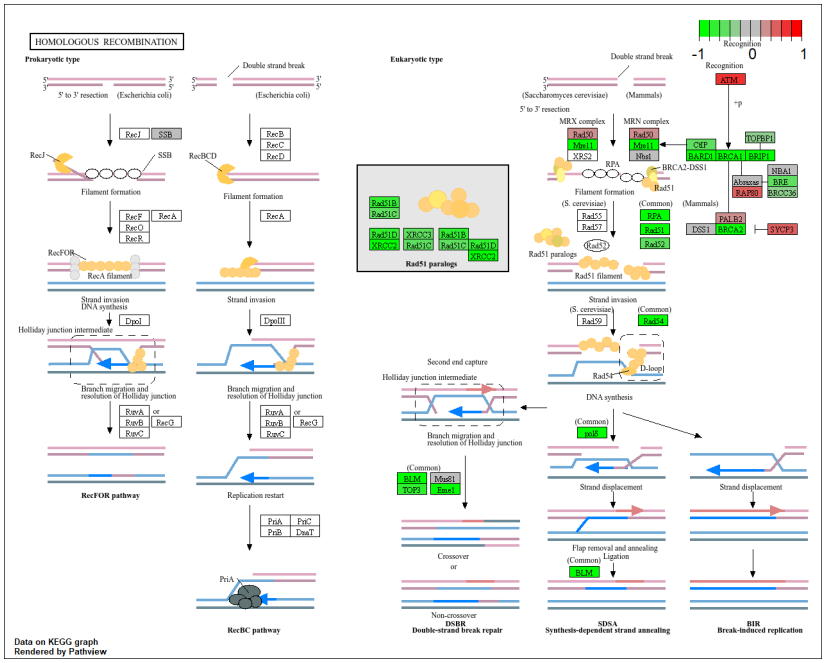



```
pathview(pathway.id = "hsa03440", gene.data = foldchanges)
```

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

Info: Working in directory C:/Users/maima/OneDrive/   /School/UCSD/Class/BIMM 143 FA'25/clas

Info: Writing image file hsa03440.pathview.png
```




## GO terms

Same analysis but this focuses on biological meaning or function of genes

```
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)
```

```
lapply(gobpres, head)
```

```
$greater
                                            p.geomean stat.mean        p.val
GO:0007156 homophilic cell adhesion       8.519724e-05  3.824205 8.519724e-05
GO:0002009 morphogenesis of an epithelium 1.396681e-04  3.653886 1.396681e-04
GO:0048729 tissue morphogenesis           1.432451e-04  3.643242 1.432451e-04
GO:0007610 behavior                       1.925222e-04  3.565432 1.925222e-04
GO:0060562 epithelial tube morphogenesis  5.932837e-04  3.261376 5.932837e-04
GO:0035295 tube development               5.953254e-04  3.253665 5.953254e-04
```

```
                                          q.val set.size         exp1
GO:0007156 homophilic cell adhesion       0.1951953      113 8.519724e-05
GO:0002009 morphogenesis of an epithelium 0.1951953      339 1.396681e-04
GO:0048729 tissue morphogenesis           0.1951953      424 1.432451e-04
GO:0007610 behavior                       0.1967577      426 1.925222e-04
GO:0060562 epithelial tube morphogenesis  0.3565320      257 5.932837e-04
GO:0035295 tube development               0.3565320      391 5.953254e-04

$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

$stats
                                         stat.mean     exp1
GO:0007156 homophilic cell adhesion       3.824205 3.824205
GO:0002009 morphogenesis of an epithelium 3.653886 3.653886
GO:0048729 tissue morphogenesis           3.643242 3.643242
GO:0007610 behavior                       3.565432 3.565432
GO:0060562 epithelial tube morphogenesis  3.261376 3.261376
GO:0035295 tube development               3.253665 3.253665
```

### Reactome

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

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

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

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

and write out to a file:

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

> Q. What pathway has the most significant "Entities p-value"? Do the most significant pathways listed match your previous KEGG results? What factors could cause differences between the two methods?

"Cell Cycle" has the most significant Entities p-value in the Reactome analysis, and this matches the most significant pathway in the KEGG results. However, the values differ because Reactome and KEGG vary in database content, statistical methods, gene-mapping strategies, and pathway definitions.

### Save our data

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