Class14 RNASeq Mini-Project

Vanesa Fernandez

```
#/ message: false
library(DESeq2)
```

Import Data

We need two "Counts" and "Metadata" (what DESeq calls colData - as it describes the columns in COUNTS)

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

metadata <- read.csv("GSE37704_metadata.csv")
```

Start with a wee peak

```
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
```

We want the column in `counts` to match the rows in the `metadata`

```
colnames(counts)
```

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

```
metadata$id
```

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

We can get of rid of the first column in `counts` to make these match

```
countData <- counts[,-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
```

```
colnames(countData) == metadata$id
```

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

```
## Are all these True?
all(c(T,T,T,T))
```

```
[1] TRUE
```

```
##and we can add it to the line above. Example:
##all(colnames(countData) == metadata$id)
## TRUE
```

Data CleanUp

Filter out zero counts

It is standard practice to remove any genes/transcripts we have no data for - i.e. zero counts in all columns. How do we do this? –> to.keep.inds function

```
to.keep.inds <- rowSums(countData) > 0
cleanCounts <- countData[to.keep.inds,]
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
```

Setup for DESeq

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

##DESeq

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

##Inspect Results

```
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>
```

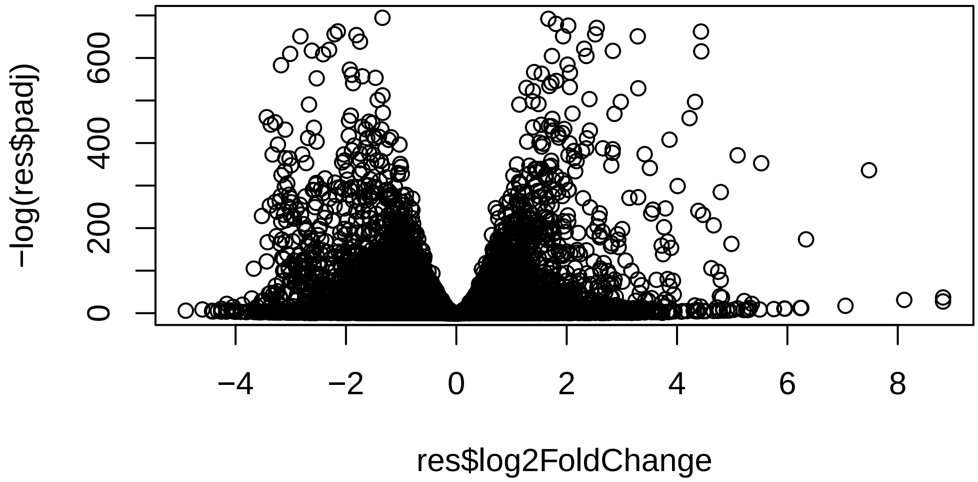
```
ENSG00000279457   29.9136      0.1792571 0.3248216   0.551863 5.81042e-01
ENSG00000187634  183.2296      0.4264571 0.1402658   3.040350 2.36304e-03
ENSG00000188976 1651.1881     -0.6927205 0.0548465 -12.630158 1.43990e-36
ENSG00000187961  209.6379      0.7297556 0.1318599   5.534326 3.12428e-08
ENSG00000187583   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
```

```
tail(cleanCounts)
```

```
                SRR493366 SRR493367 SRR493368 SRR493369 SRR493370 SRR493371
ENSG00000278198         0         3         0         1         1         0
ENSG00000273748        23        28        26        45        40        54
ENSG00000278817         3         1         4         1         2         4
ENSG00000278384         0         1         2         1         2         1
ENSG00000276345        72        73        91        55        67        87
ENSG00000271254       188       211       222       148       150       161
```

Data Viz

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



Pathway Analysis

##Annotattion of genes 1st translate Ensemble IDs in `res` object to Entrez and gen symbol formats

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

```r
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"
```

Let’s map “SYMBOL”, “ENTREZID”, “GENENAME” from our “ENSEMBL” ids

```r
##book-keeping
res$genename <- mapIds(org.Hs.eg.db,
                 keys= rownames(res),
                 keytype = "ENSEMBL",
                 column = "GENENAME")

res$symbol <- mapIds(org.Hs.eg.db,
                 keys= rownames(res),
                 keytype = "ENSEMBL",
                 column = "SYMBOL")

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

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               genename      symbol      entrez
                  <numeric>            <character> <character> <character>
ENSG00000279457 6.86555e-01                     NA          NA          NA
ENSG00000187634 5.15718e-03 sterile alpha motif ..      SAMD11      148398
ENSG00000188976 1.76549e-35 NOC2 like nucleolar ..       NOC2L       26155
ENSG00000187961 1.13413e-07 kelch like family me..      KLHL17      339451
ENSG00000187583 9.19031e-01 pleckstrin homology ..     PLEKHN1       84069
ENSG00000187642 4.03379e-01 PPARGC1 and ESRR ind..       PERM1       84808
```

Before moving on, let's focus in on a subset of "top" hits We can use as a starting point log2FC of +2/-2 and adjusted p-value of less than 0.05. This is a way to start nailing down the data.

```r
top.inds <- (abs(res$log2FoldChange) > 2) & (res$padj < 0.05)
top.inds[is.na(top.inds)] <- FALSE
```

Let's save our top genes to a CSV file…

```r
top.genes <- res[top.inds,]
write.csv(top.genes, file = "top_geneset.csv")
```

Now we can do some pathway analysis

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

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

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

The **gage** function wants a vector of importance as input with gene names as labels - KEGG speaks Entrez

```r
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
```

```r
keggres <- gage(foldchanges, gsets=kegg.sets.hs)
```

```r
attributes(keggres)
```

```
$names
[1] "greater" "less"    "stats"
```

Different KEGG pathways overlapping

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

```
                                             p.geomean stat.mean        p.val
hsa04110 Cell cycle                       8.995727e-06 -4.378644 8.995727e-06
hsa03030 DNA replication                  9.424076e-05 -3.951803 9.424076e-05
hsa03013 RNA transport                    1.246882e-03 -3.059466 1.246882e-03
hsa03440 Homologous recombination         3.066756e-03 -2.852899 3.066756e-03
hsa04114 Oocyte meiosis                   3.784520e-03 -2.698128 3.784520e-03
hsa00010 Glycolysis / Gluconeogenesis 8.961413e-03 -2.405398 8.961413e-03
                                                q.val set.size         exp1
hsa04110 Cell cycle                       0.001448312      121 8.995727e-06
hsa03030 DNA replication                  0.007586381       36 9.424076e-05
hsa03013 RNA transport                    0.066915974      144 1.246882e-03
hsa03440 Homologous recombination         0.121861535       28 3.066756e-03
hsa04114 Oocyte meiosis                   0.121861535      102 3.784520e-03
hsa00010 Glycolysis / Gluconeogenesis 0.212222694       53 8.961413e-03
```

hsa04110 Cell cycle

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

GO Gene Ontology

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

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

gores <- gage(foldchanges, gsets=gobpsets)
```

```
head(gores$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 Analysis – building a website-like for better vis of results and even has been used in papers. ##https://reactome.org/user/guide

To run it online, we need to make a text file with a gene id per line

```
sig_genes <- res[res$padj <= 0.05 & !is.na(res$padj), "symbol"]
print(paste("Total number of significant genes:", length(sig_genes)))
```

```
[1] "Total number of significant genes: 8147"
```

```
invisible(sig_genes)
```

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

Now we can take now the generated ("significant_genes.txt") file and upload it to: https://reactome.org/PathwayBrowser/#TOOL=AT)

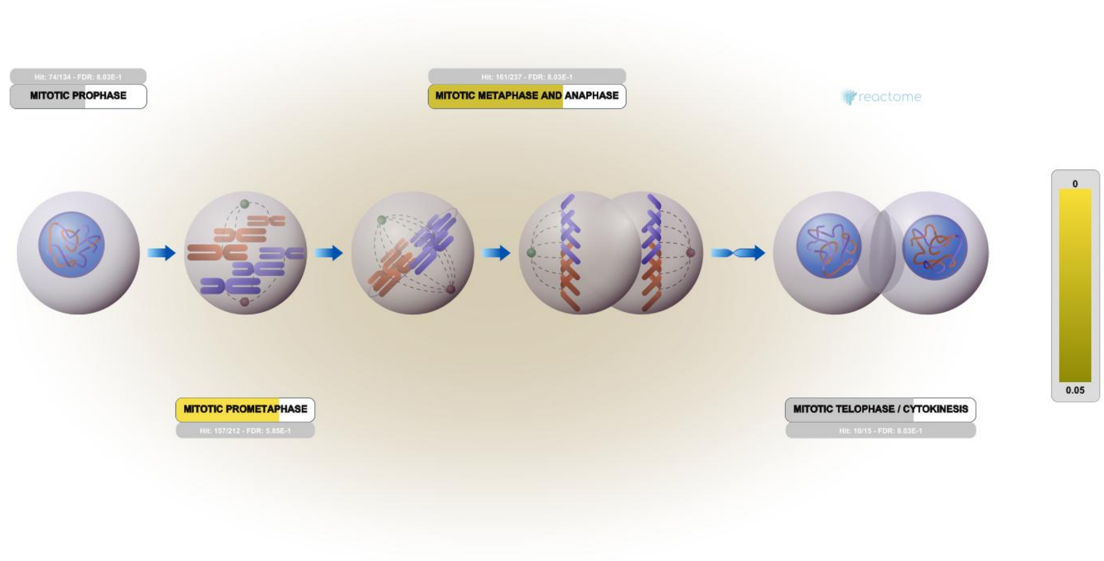

Hit: 74/134 - FDR: 8.03E-1
MITOTIC PROPHASE
Hit: 161/237 - FDR: 8.03E-1
MITOTIC METAPHASE AND ANAPHASE
reactome
0
0.05
MITOTIC PROMETAPHASE
Hit: 157/212 - FDR: 5.85E-1
MITOTIC TELOPHASE / CYTOKINESIS
Hit: 10/15 - FDR: 8.03E-1

Figure 1: Taken from reactome, this is Mitotic Phase of my gene expression

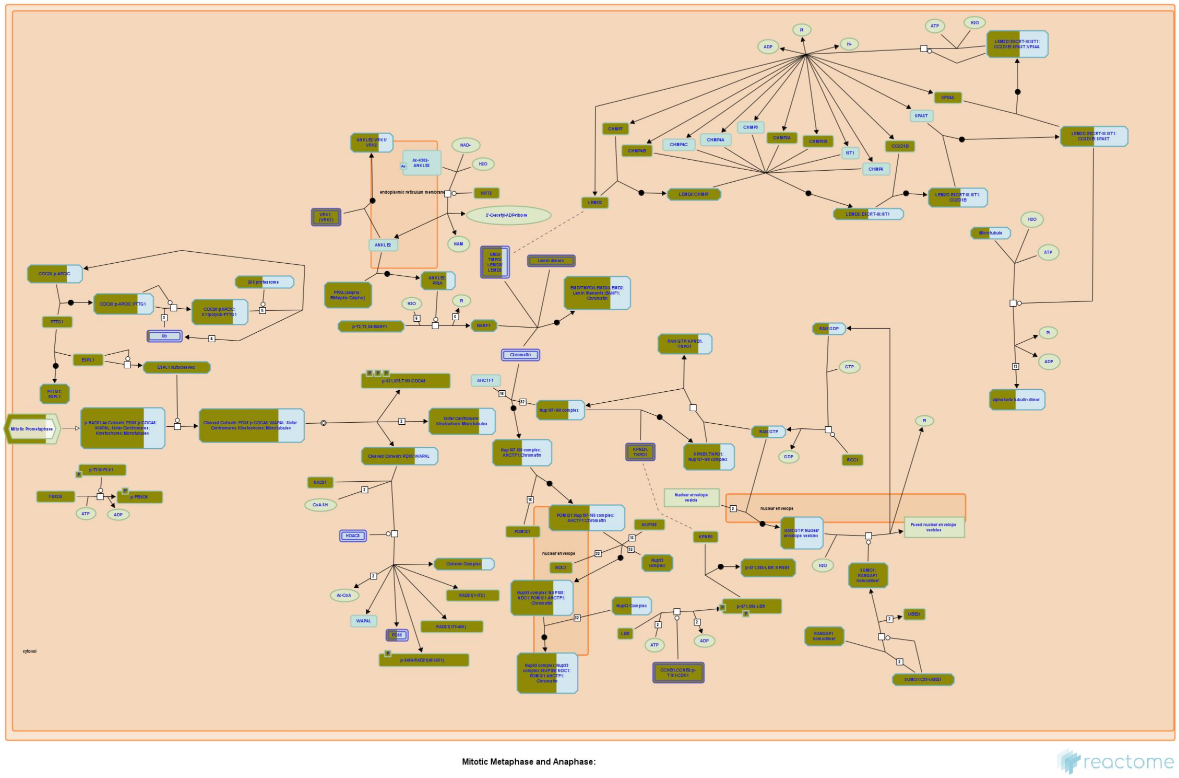


Figure 2: Here, we can visualize the mitotic pathway in a tree way to visualize the gene intereaction. This is coming from our data