Class 14: RNASeq Mini Project

Sawyer Randles PID: A69034741

Import Data

We need two things "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')
```

Data CleanUp

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
```

```
head(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
```

We want the columns in the `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 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
```

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

```
[1] TRUE
```

Filter out zero counts

It is standard practice to remove any genes/transcripts that we have no data for - i.e. zero counts in all columns.

```
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 DESeup

```
library(DESeq2)
```

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

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

DESeq

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

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship

final dispersion estimates

fitting model and testing
```

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

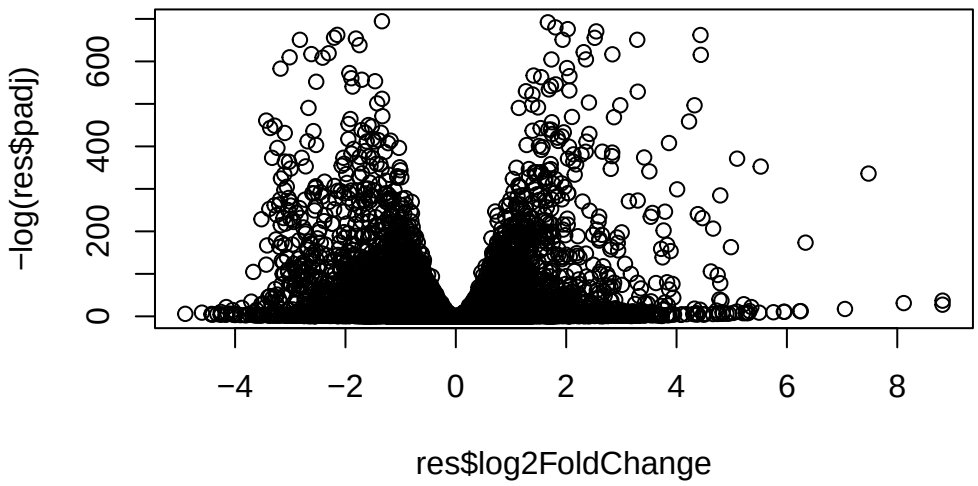
Inspect Results

```
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
```

Data Viz

```
plot(x = res$log2FoldChange, y = -log(res$padj))
```



Annotation of genes

First I need to translate my Ensemble IDs in my `res` object to Entrez and gene symbol formats.

For this I will use the AnnotationDbi package and it's `mapIds()` function.

```
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"
```

Let's map to "SYMBOL", "ENTREZID" "GENENAME" from "ENSEMBL" ids.

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

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

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

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

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

```
'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               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 going any further lets focus in on a subset of "top" hits.

We can use as a starting point log2FC of +2/-2 and an adjusted P-value of 0.05.

```
# when there is no chance of passing, the program does not even test p-vlaues that are alread
# you can go ahead and make these false in your selection criteria so they are excluded
top.inds <- (abs(res$log2FoldChange) > 2) & (abs(res$padj) > 0.05)
top.inds[is.na(top.inds)] <- FALSE
```

Let's save our "top genes" to a CSV file...

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

Pathway Analysis

Now we can do some pathway analysis

```
library(gage)
```

```
library(gageData)
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).
##############################################################################
```

```
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)
```

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

```
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
```

Run gage with these values

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

```
attributes(keggres)
```

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

```
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 0.065461279
hsa03440 Homologous recombination                 3.066756e-03 0.128803765
hsa04114 Oocyte meiosis                           3.784520e-03 0.132458191
                                                  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(foldchanges, pathway.id = "hsa04110")
```

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

```
Info: Working in directory /Users/sawyerrandles/Documents/BGGN213 Bioinformatics F24/BGGN213
```

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

```
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
```

```
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
```

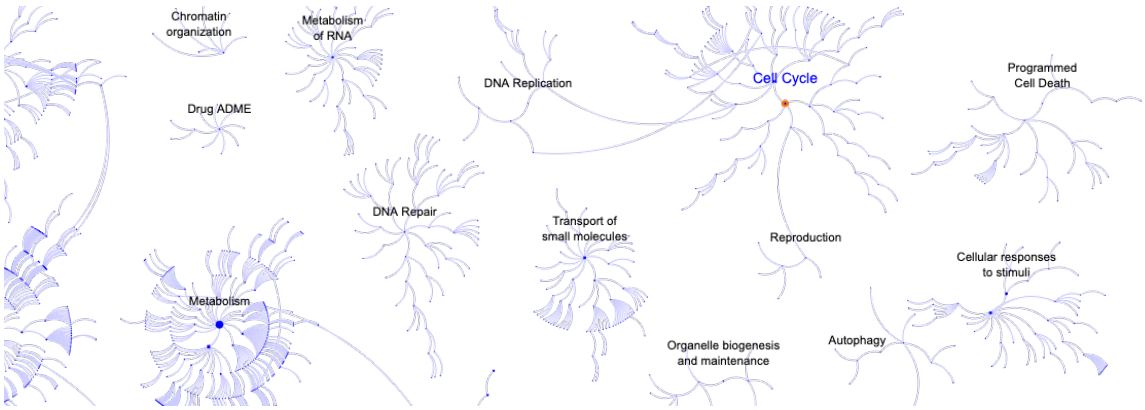
To run reactome online we need to make a little text file with a gene id per line.

```
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"
```

```
#sig_genes
```

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




```
#all() checks to see if all conditions are true
x <- all(c(T,T,F,T))

{if (all(x)) {
  cat("me happy")
} else {}
  cat("me no happy")
  }
```

```
me no happy
```

```
c(T,T,F,T) & c(F,T,T,T)
```

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