

Class13

Gregory Jordan

About our Input Data

The data for for hands-on session comes from GEO entry: GSE37704, which is associated with the following publication:

Trapnell C, Hendrickson DG, Sauvageau M, Goff L et al. “Differential analysis of gene regulation at transcript resolution with RNA-seq”. Nat Biotechnol 2013 Jan;31(1):46-53. PMID: 23222703

1. Differential Expression Analysis

```
#load in DESeq2
library(DESeq2)
```

```
metaFile <- "https://bioboot.github.io/bimm143_W18/class-material/GSE37704_metadata.csv"
countFile <- "https://bioboot.github.io/bimm143_W18/class-material/GSE37704_featurecounts.csv"
```

```
#import the metadata and check it out
```

```
colData = read.csv(metaFile,row.names = 1)
```

```
head(colData)
```

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

```
SRR493370      hoxa1_kd
SRR493371      hoxa1_kd
```

```
# Import countdata
countData = read.csv(countFile, row.names=1)
head(countData)
```

```
      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
```

need to remove our length column to make our count identifiers line up with metadata

Q1. Complete the code below to remove the troublesome first column from count-Data

Note we need to remove the odd first \$length col

```
countData <- as.matrix(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
```

Test that our count data is in the right order with metadata now

```
#remember that all will test if every value in the resulting vector (comparing each element)
all(colnames(countData) == rownames(colData))
```

```
[1] TRUE
```

Lots of 0 counts still here and it is good practice to remove zeroes before we go through other things because 0s will mess with our statistical tests

Q.2 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).

Tip: What will rowSums() of countData return and how could you use it in this context?

```
# Filter count data where you have 0 read count across all samples.
#keep.inds is a cool feature to keep indices based on a condition
keep.inds <- rowSums(countData) != 0
counts <- countData[keep.inds,]
```

```
nrow(countData)
```

```
[1] 19808
```

```
nrow(counts)
```

```
[1] 15975
```

Running DESeq2

```
#already loaded ddseq at the beginning
#tilda condition for the experiment design
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

```
dds <- DESeq(dds)

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

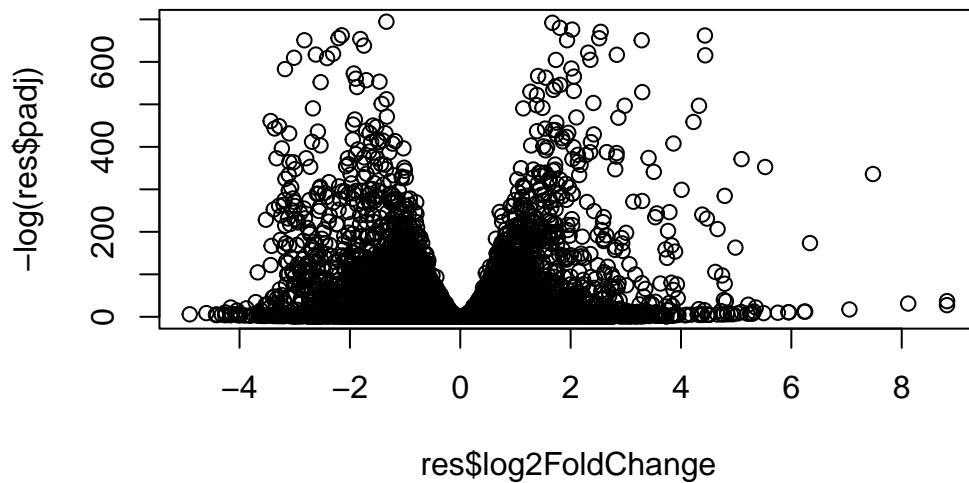
Q3. Call the `summary()` function on your results to get a sense of how many genes are up or down-regulated at the default 0.1 p-value cutoff.

```
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
```

```
#make volcano plot
plot(res$log2FoldChange, -log(res$padj) )
```

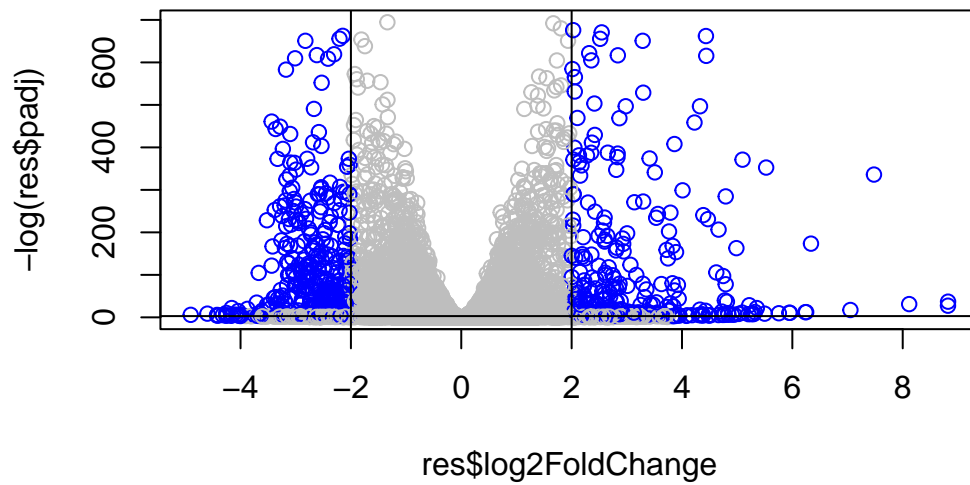


Q4. Improve this plot by completing the below code, which adds color and axis labels

```
# Make a color vector for all genes
mycols <- rep("gray", nrow(counts))
mycols[res$log2FoldChange > 2] <- "blue"
mycols[res$log2FoldChange < -2] <- "blue"
mycols[res$padj > 0.05] <- "gray"

plot(res$log2FoldChange, -log(res$padj), col=mycols)

abline(v=c(-2,+2))
abline(h=-log(0.05))
```



Adding Gene Annotation

Q5. Use the `mapIDs()` function multiple times to add **SYMBOL**, **ENTREZID** and **GENENAME** annotation to our results by completing the code below.

```
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"
```

```
#x = the annotationDb object
#keys = the query essentially
#column = what type you want to convert the keys into. essentially your key goes into the
#multivals = when there are multiple values that can be returned, return the first value

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, 10)
```

log2 fold change (MLE): condition hoxa1 kd vs control sirna

Wald test p-value: condition hoxa1 kd vs control sirna

DataFrame with 10 rows and 9 columns

	baseMean	log2FoldChange	lfcSE	stat	pvalue
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
ENSG00000279457	29.913579	0.1792571	0.3248216	0.551863	5.81042e-01

ENSG00000187634	183.229650	0.4264571	0.1402658	3.040350	2.36304e-03
ENSG00000188976	1651.188076	-0.6927205	0.0548465	-12.630158	1.43990e-36
ENSG00000187961	209.637938	0.7297556	0.1318599	5.534326	3.12428e-08
ENSG00000187583	47.255123	0.0405765	0.2718928	0.149237	8.81366e-01
ENSG00000187642	11.979750	0.5428105	0.5215598	1.040744	2.97994e-01
ENSG00000188290	108.922128	2.0570638	0.1969053	10.446970	1.51282e-25
ENSG00000187608	350.716868	0.2573837	0.1027266	2.505522	1.22271e-02
ENSG00000188157	9128.439422	0.3899088	0.0467163	8.346304	7.04321e-17
ENSG00000237330	0.158192	0.7859552	4.0804729	0.192614	8.47261e-01
	padj	symbol	entrez		name
	<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 ..	
ENSG00000187642	4.03379e-01	PERM1	84808	PPARGC1 and ESRR ind..	
ENSG00000188290	1.30538e-24	HES4	57801	hes family bHLH tran..	
ENSG00000187608	2.37452e-02	ISG15	9636	ISG15 ubiquitin like..	
ENSG00000188157	4.21963e-16	AGRN	375790		agrin
ENSG00000237330	NA	RNF223	401934	ring finger protein ..	

Q6. Finally for this section let's reorder these results by adjusted p-value and save them to a CSV file in your current project directory.

```
res.ordered <- res[order(res$padj),]
head(res.ordered)
```

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>
ENSG00000117519	4483.63	-2.42272	0.0600016	-40.3776	0
ENSG00000183508	2053.88	3.20196	0.0724172	44.2154	0
ENSG00000159176	5692.46	-2.31374	0.0575534	-40.2016	0
ENSG00000150938	7442.99	-2.05963	0.0538449	-38.2512	0
ENSG00000116016	4423.95	-1.88802	0.0431680	-43.7366	0
ENSG00000136068	3796.13	-1.64979	0.0439354	-37.5504	0
	padj	symbol	entrez		name
	<numeric>	<character>	<character>		<character>
ENSG00000117519	0	CNN3	1266		calponin 3

ENSG00000183508	0	TENT5C	54855	terminal nucleotidyl..
ENSG00000159176	0	CSRP1	1465	cysteine and glycine..
ENSG00000150938	0	CRIM1	51232	cysteine rich transm..
ENSG00000116016	0	EPAS1	2034	endothelial PAS doma..
ENSG00000136068	0	FLNB	2317	filamin B

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

QC with PCA

the `prcomp()` function in base R is often used to check principal components contributing to the variability in the counts

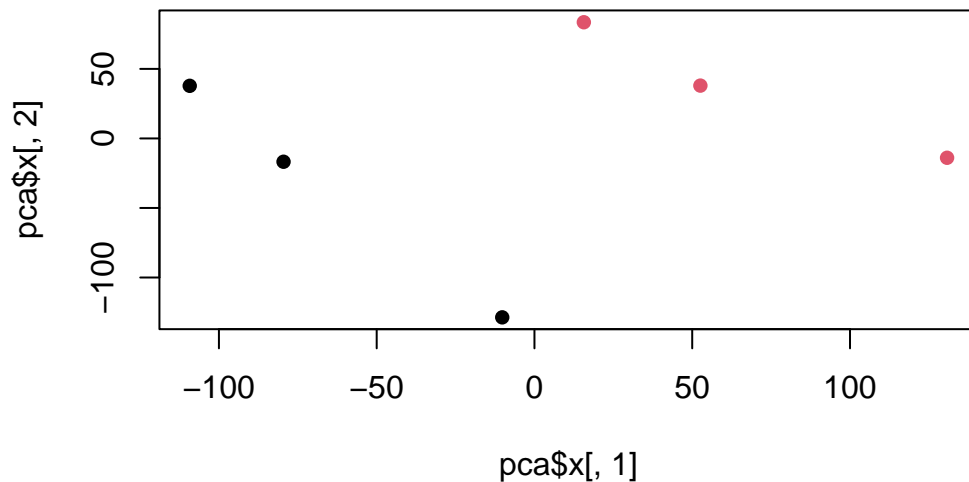
```
pca <- prcomp(t(counts), scale=TRUE)
summary(pca)
```

Importance of components:

	PC1	PC2	PC3	PC4	PC5	PC6
Standard deviation	87.7211	73.3196	32.89604	31.15094	29.18417	6.648e-13
Proportion of Variance	0.4817	0.3365	0.06774	0.06074	0.05332	0.000e+00
Cumulative Proportion	0.4817	0.8182	0.88594	0.94668	1.00000	1.000e+00

Our PCA score plot (aka PC1 vs PC2)

```
plot(pca$x[,1], pca$x[,2], col = as.factor(colData$condition), pch=16)
```



#this is good. we see the major variance in the dataset being found by PCA is consistent w

2. Pathway Analysis

we can use `gage()` with KEGG and GO

```
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)
```

what `gage()` wants as input is that vector of importance - in our case that will be the log2 fold-change values. this vector should have `names()` that are entrez IDs

```
foldchange <- res$log2FoldChange
names(foldchange) <- res$entrez
#now we have fold change named with entrez ids
```

```
head(foldchange)
```

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

```
#focus on signaling and metabolic pathways only
```

```
kegg.sets.hs = kegg.sets.hs[sigmet.idx.hs]
```

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

```
attributes(keggres)
```

```
$names
```

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

```
head(keggres$less, 5)
```

```

                                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
```

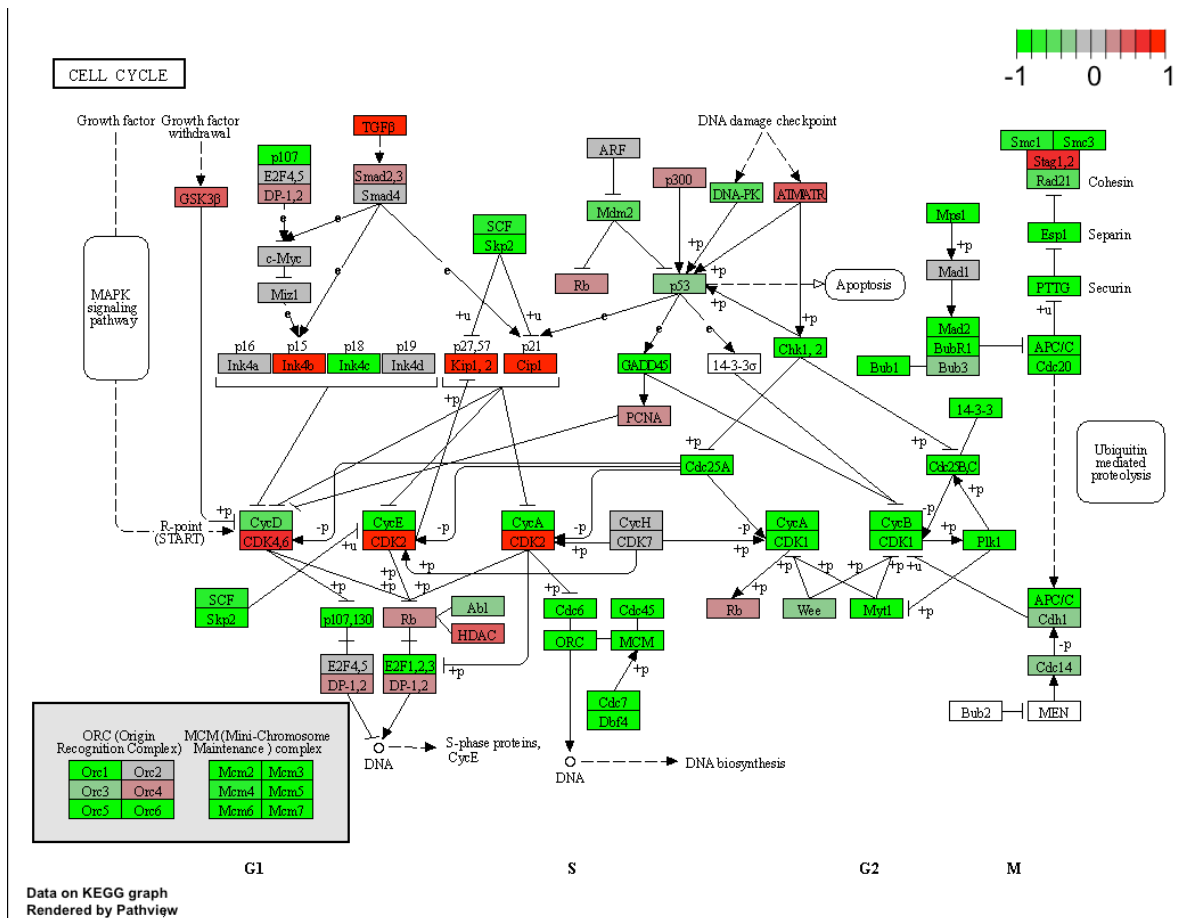
hsa03013	RNA transport	1.246882e-03	-3.059466	1.246882e-03
hsa03440	Homologous recombination	3.066756e-03	-2.852899	3.066756e-03
hsa04114	Oocyte meiosis	3.784520e-03	-2.698128	3.784520e-03
		q.val	set.size	exp1
hsa04110	Cell cycle	0.001448312	121	8.995727e-06
hsa03030	DNA replication	0.007586381	36	9.424076e-05
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

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

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

Info: Working in directory /Users/gregoryjordan/Desktop/BGGN213/BGGN 213_R Project/Class13

Info: Writing image file hsa04110.pathview.png



3. 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]
```

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

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

```
$greater
```

	p.geomean	stat.mean	p.val
G0:0007156 homophilic cell adhesion	8.519724e-05	3.824205	8.519724e-05
G0:0002009 morphogenesis of an epithelium	1.396681e-04	3.653886	1.396681e-04
G0:0048729 tissue morphogenesis	1.432451e-04	3.643242	1.432451e-04

G0:0007610	behavior	2.195494e-04	3.530241	2.195494e-04
G0:0060562	epithelial tube morphogenesis	5.932837e-04	3.261376	5.932837e-04
G0:0035295	tube development	5.953254e-04	3.253665	5.953254e-04

		q.val	set.size	exp1
G0:0007156	homophilic cell adhesion	0.1951953	113	8.519724e-05
G0:0002009	morphogenesis of an epithelium	0.1951953	339	1.396681e-04
G0:0048729	tissue morphogenesis	0.1951953	424	1.432451e-04
G0:0007610	behavior	0.2243795	427	2.195494e-04
G0:0060562	epithelial tube morphogenesis	0.3711390	257	5.932837e-04
G0:0035295	tube development	0.3711390	391	5.953254e-04

\$less

		p.geomean	stat.mean	p.val
G0:0048285	organelle fission	1.536227e-15	-8.063910	1.536227e-15
G0:0000280	nuclear division	4.286961e-15	-7.939217	4.286961e-15
G0:0007067	mitosis	4.286961e-15	-7.939217	4.286961e-15
G0:0000087	M phase of mitotic cell cycle	1.169934e-14	-7.797496	1.169934e-14
G0:0007059	chromosome segregation	2.028624e-11	-6.878340	2.028624e-11
G0:0000236	mitotic prometaphase	1.729553e-10	-6.695966	1.729553e-10

		q.val	set.size	exp1
G0:0048285	organelle fission	5.841698e-12	376	1.536227e-15
G0:0000280	nuclear division	5.841698e-12	352	4.286961e-15
G0:0007067	mitosis	5.841698e-12	352	4.286961e-15
G0:0000087	M phase of mitotic cell cycle	1.195672e-11	362	1.169934e-14
G0:0007059	chromosome segregation	1.658603e-08	142	2.028624e-11
G0:0000236	mitotic prometaphase	1.178402e-07	84	1.729553e-10

\$stats

		stat.mean	exp1
G0:0007156	homophilic cell adhesion	3.824205	3.824205
G0:0002009	morphogenesis of an epithelium	3.653886	3.653886
G0:0048729	tissue morphogenesis	3.643242	3.643242
G0:0007610	behavior	3.530241	3.530241
G0:0060562	epithelial tube morphogenesis	3.261376	3.261376
G0:0035295	tube development	3.253665	3.253665

4. Reactome Analysis

```
#get your significant genes
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"
```

```
#make a table of your significant genes
write.table(sig_genes, file="significant_genes.txt", row.names=FALSE, col.names=FALSE, quo
```

Q7: 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?

Endosomal/Vacuolar pathway is the most significant pathway for KEGG

detection of chemical stimulus involved in sensory perception for GO analysis

These results do not match. I imagine differences could be caused by different search algorithms by the softwares, among different cutoffs used by the different softwares.