

# Class\_14\_RNA\_seq\_mini\_project

Johann Tailor

## Data Import and Reading

```
library(DESeq2)
```

Loading required package: S4Vectors

Loading required package: stats4

Loading required package: BiocGenerics

Attaching package: 'BiocGenerics'

The following objects are masked from 'package:stats':

IQR, mad, sd, var, xtabs

The following objects are masked from 'package:base':

anyDuplicated, aperm, append, as.data.frame, basename, cbind,  
colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,  
get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,  
match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,  
Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,  
table, tapply, union, unique, unsplit, which.max, which.min

Attaching package: 'S4Vectors'

The following object is masked from 'package:utils':

findMatches

The following objects are masked from 'package:base':

expand.grid, I, unname

Loading required package: IRanges

Loading required package: GenomicRanges

Loading required package: GenomeInfoDb

Loading required package: SummarizedExperiment

Loading required package: MatrixGenerics

Loading required package: matrixStats

Attaching package: 'MatrixGenerics'

The following objects are masked from 'package:matrixStats':

colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,  
colCounts, colCummaxs, colCummins, colCumprods, colCumsums,  
colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,  
colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,  
colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,  
colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,  
colWeightedMeans, colWeightedMedians, colWeightedSds,  
colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,  
rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,  
rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,  
rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,  
rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,  
rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,  
rowWeightedMads, rowWeightedMeans, rowWeightedMedians,  
rowWeightedSds, rowWeightedVars

Loading required package: Biobase

Welcome to Bioconductor

Vignettes contain introductory material; view with  
'browseVignettes()'. To cite Bioconductor, see  
'citation("Biobase")', and for packages 'citation("pkgname")'.

Attaching package: 'Biobase'

The following object is masked from 'package:MatrixGenerics':

rowMedians

The following objects are masked from 'package:matrixStats':

anyMissing, rowMedians

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

## Data Exploration

```
head(metaFile)
```

	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

```
head(countFile)
```

	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

## Check for similarity between column names:

```
colnames(countFile)
```

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

```
head(metaFile)
```

	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

## Filter out the zeros

removing the length column

```
countdata <- countFile[, -1]
```

Checking for the sample exactness

```
#Using == to check if they are perfectly same
colnames(countdata) == metaFile$id
```

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

Some genes have no values, so we want to get rid of it.

```
#Only select for values which have a sum greater than 0 for each row/gene:
non.zero.inds <- rowSums(countdata) > 0

#Looks for that in all the rows/genes:
non.zero.counts <- countdata[non.zero.inds,]

head(non.zero.counts)
```

	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

```
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

## Setup for DESeq

```
library(DESeq2)
```

```
#DESeq analysis
```

```
dds <- DESeqDataSetFromMatrix(countData = non.zero.counts,  
                              colData = metaFile,  
                              design = ~condition)
```

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

```
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)
```

```
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>

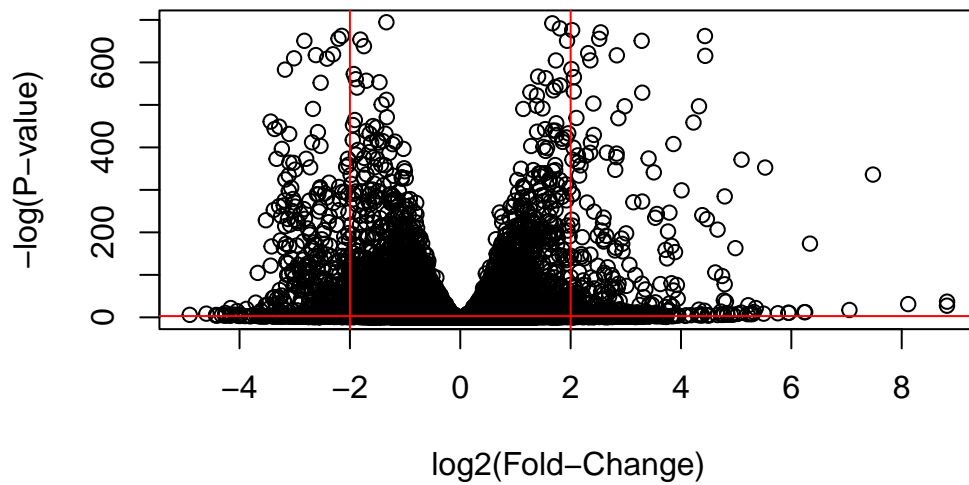
ENSG00000279457	29.9136	0.1792571	0.3248216	0.551863	5.81042e-01
ENSG00000187634	183.2296	0.4264571	0.1402658	3.040350	2.36304e-03
ENSG00000188976	1651.1881	-0.6927205	0.0548465	-12.630158	1.43989e-36
ENSG00000187961	209.6379	0.7297556	0.1318599	5.534326	3.12428e-08
ENSG00000187583	47.2551	0.0405765	0.2718928	0.149237	8.81366e-01
ENSG00000187642	11.9798	0.5428105	0.5215599	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

##Making the Volcano Plot:

```
plot(res$log2FoldChange, -log(res$padj),
     ylab="-log(P-value)",
     xlab="log2(Fold-Change)")
abline(v=2, col= "red")
abline(v=-2, col= "red")
abline(h=-log(0.05), col= "red")
```



We can also use ggplot but first we have to make it as a data frame:

```
res_df <- as.data.frame(res)
```

```
library(ggplot2)
```

```
#Color all points gray:
```

```
mycols <- rep("gray", nrow(res_df))
```

```
#Setting limits:
```

```
mycols
```

```
[1] "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray"
[11] "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray"
[21] "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray"
[31] "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray"
[41] "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray"
[51] "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray"
[61] "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray"
[71] "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray"
[81] "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray" "gray"
```



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

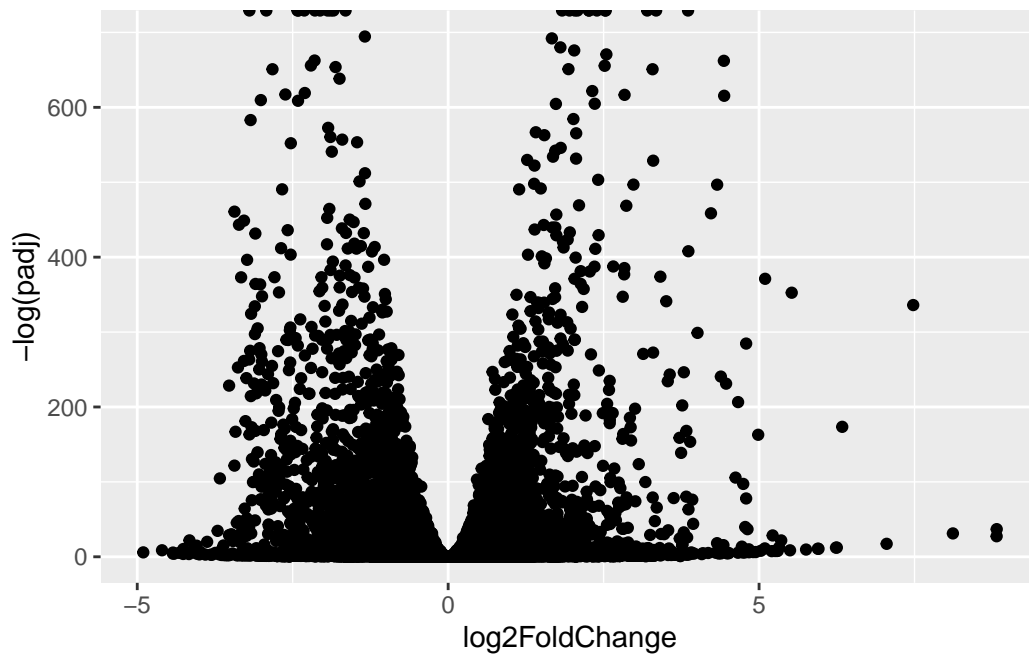
[illegible]

[illegible]

[illegible]

```
ggplot(res_df, aes(x = log2FoldChange, y = -log(padj))) +  
  geom_point()
```

Warning: Removed 1237 rows containing missing values (`geom\_point()`).



## Annotating the genes

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

Let's see the databases that we can translate between

```
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 try to use different databases and translate the IDs:

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

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

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

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

```
head(res_df$entrez)
```

```
[1] NA      "148398" "26155"  "339451" "84069"  "84808"
```

#Result extraction

## KEGG and GO analysis

```
BiocManager::install( c("pathview", "gage", "gageData"))
```

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

```
foldchanges <- res_df$log2FoldChange
names(foldchanges) <- res_df$entrez
head(foldchanges)
```

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

## Pathway Analysis

```
library(gage)
library(gageData)

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"

```

Let's look at what is inside:

```

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

```

```

[1] 0.17925708 0.42645712 -0.69272046 0.72975561 0.04057653 0.54281049

```

```

# Get the results
keggres = gage(foldchanges, gsets=kegg.sets.hs)

```

```

head(keggres$less)

```

	p.geomean	stat.mean	p.val	q.val
hsa00232 Caffeine metabolism	NA	NaN	NA	NA
hsa00983 Drug metabolism - other enzymes	NA	NaN	NA	NA
hsa00230 Purine metabolism	NA	NaN	NA	NA
hsa04514 Cell adhesion molecules (CAMs)	NA	NaN	NA	NA
hsa04010 MAPK signaling pathway	NA	NaN	NA	NA
hsa04012 ErbB signaling pathway	NA	NaN	NA	NA

	set.size	expl
hsa00232 Caffeine metabolism	0	NA
hsa00983 Drug metabolism - other enzymes	0	NA

hsa00230	Purine metabolism	0	NA
hsa04514	Cell adhesion molecules (CAMs)	0	NA
hsa04010	MAPK signaling pathway	0	NA
hsa04012	ErbB signaling pathway	0	NA

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

Warning: None of the genes or compounds mapped to the pathway!  
Argument gene.idtype or cpd.idtype may be wrong.

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

Info: Working in directory /Users/johanntailor/Desktop/UCSD Courses/Winter 2024/Bioinformati

Info: Writing image file hsa04110.pathview.png

## Gene Ontology (GO)

```
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)
```

		p.geomean	stat.mean	p.val	q.val
\$greater					
GO:0000002	mitochondrial genome maintenance	NA	NaN	NA	NA
GO:0000003	reproduction	NA	NaN	NA	NA
GO:0000012	single strand break repair	NA	NaN	NA	NA
GO:0000018	regulation of DNA recombination	NA	NaN	NA	NA
GO:0000019	regulation of mitotic recombination	NA	NaN	NA	NA
GO:0000022	mitotic spindle elongation	NA	NaN	NA	NA

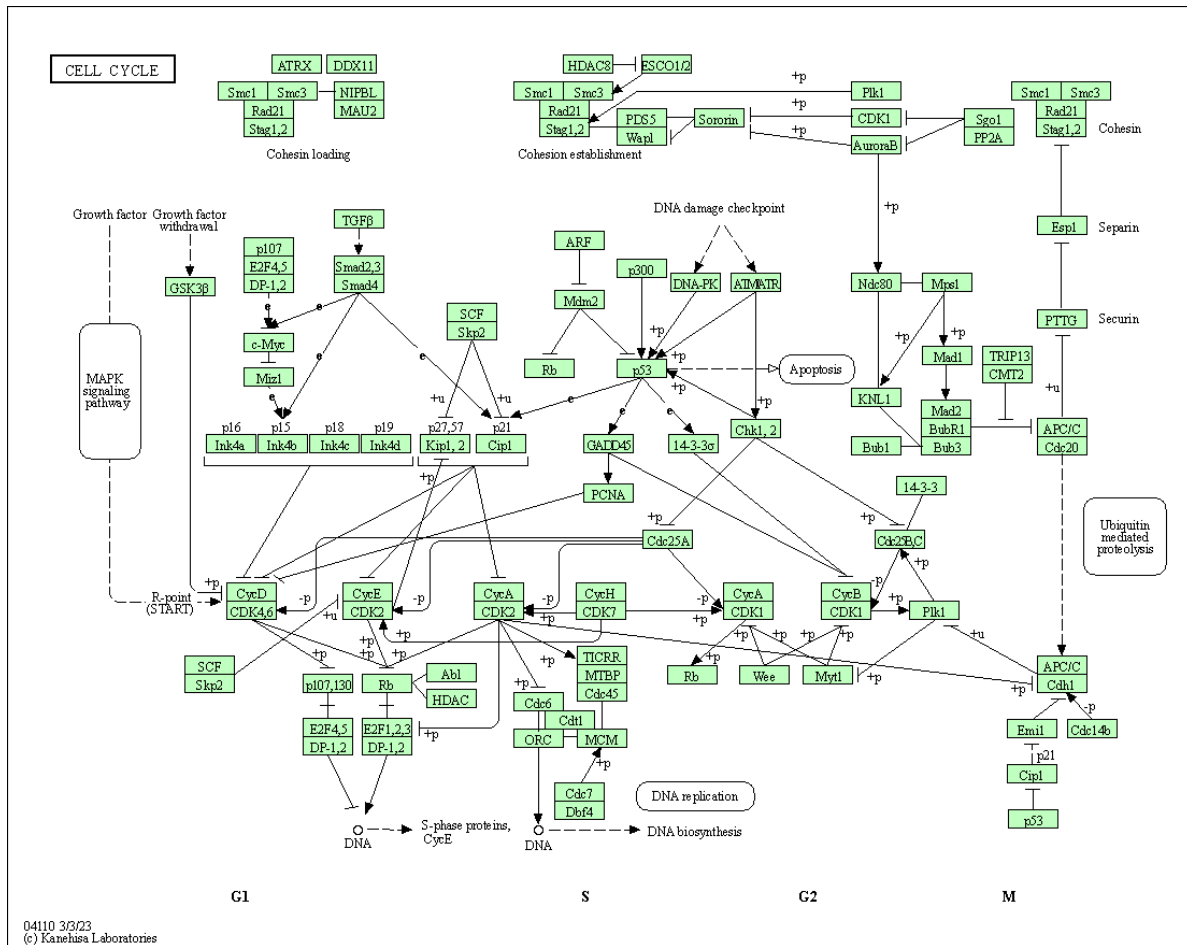


Figure 1: Cell cycle pathway from KEGG

	set.size	exp1
G0:0000002 mitochondrial genome maintenance	0	NA
G0:0000003 reproduction	0	NA
G0:0000012 single strand break repair	0	NA
G0:0000018 regulation of DNA recombination	0	NA
G0:0000019 regulation of mitotic recombination	0	NA
G0:0000022 mitotic spindle elongation	0	NA

\$less

	p.geomean	stat.mean	p.val	q.val
G0:0000002 mitochondrial genome maintenance	NA	NaN	NA	NA
G0:0000003 reproduction	NA	NaN	NA	NA
G0:0000012 single strand break repair	NA	NaN	NA	NA
G0:0000018 regulation of DNA recombination	NA	NaN	NA	NA
G0:0000019 regulation of mitotic recombination	NA	NaN	NA	NA
G0:0000022 mitotic spindle elongation	NA	NaN	NA	NA

	set.size	exp1
G0:0000002 mitochondrial genome maintenance	0	NA
G0:0000003 reproduction	0	NA
G0:0000012 single strand break repair	0	NA
G0:0000018 regulation of DNA recombination	0	NA
G0:0000019 regulation of mitotic recombination	0	NA
G0:0000022 mitotic spindle elongation	0	NA

\$stats

	stat.mean	exp1
G0:0000002 mitochondrial genome maintenance	NaN	NA
G0:0000003 reproduction	NaN	NA
G0:0000012 single strand break repair	NaN	NA
G0:0000018 regulation of DNA recombination	NaN	NA
G0:0000019 regulation of mitotic recombination	NaN	NA
G0:0000022 mitotic spindle elongation	NaN	NA

```
head(gobpres$less)
```

	p.geomean	stat.mean	p.val	q.val
G0:0000002 mitochondrial genome maintenance	NA	NaN	NA	NA
G0:0000003 reproduction	NA	NaN	NA	NA
G0:0000012 single strand break repair	NA	NaN	NA	NA
G0:0000018 regulation of DNA recombination	NA	NaN	NA	NA
G0:0000019 regulation of mitotic recombination	NA	NaN	NA	NA

	set.size	exp1	NA	NaN	NA	NA
G0:0000022 mitotic spindle elongation	0	NA				
G0:0000002 mitochondrial genome maintenance	0	NA				
G0:0000003 reproduction	0	NA				
G0:0000012 single strand break repair	0	NA				
G0:0000018 regulation of DNA recombination	0	NA				
G0:0000019 regulation of mitotic recombination	0	NA				
G0:0000022 mitotic spindle elongation	0	NA				

## Reactome Analysis

```
#Pick all genes with fold change greater than 2

#Checks two things at once:

indices <- (abs(res_df$log2FoldChange) > 2) & (abs(res_df$padj) < 0.05)

my_genes <- res_df$symbol[indices]

cat(head(my_genes), sep = "\n")
```

```
HES4
HES2
DRAXIN
CDA
RUNX3
AUNIP
```

```
write.table(my_genes, file="my_genes.txt", quote = FALSE, row.names = FALSE, col.names = F
```

Added the file to

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

