class15 mini project

Abraham Rachlin

```
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, saveRDS, setdiff,
    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
```

```r
metaFile <- "GSE37704_metadata.csv"
countFile <- "GSE37704_featurecounts.csv"

# Import metadata and take a peak
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
```

```r
# 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
```

Q. Complete the code below to remove the troublesome first column from countData

```
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
```

Q. 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).

```
to.keep <- rowSums(countData[,1:2] == 0) == 0
sum(to.keep)
```

```
[1] 13761
```

```
countData = countData[to.keep, ]
head(countData)
```

```
                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
```

```
dds = DESeqDataSetFromMatrix(countData = countData,
                             colData=colData,
                             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
```

```
dds
```

```
class: DESeqDataSet
dim: 13761 6
metadata(1): version
assays(4): counts mu H cooks
rownames(13761): ENSG00000279457 ENSG00000187634 ... ENSG00000276345
  ENSG00000271254
rowData names(22): baseMean baseVar ... deviance maxCooks
colnames(6): SRR493366 SRR493367 ... SRR493370 SRR493371
colData names(2): condition sizeFactor
```

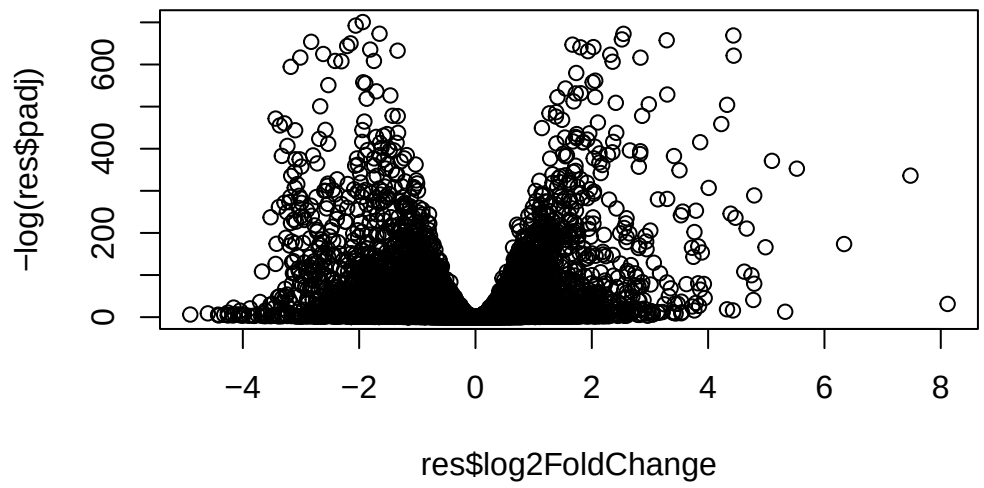
```
res = results(dds, contrast=c("condition", "hoxa1_kd", "control_sirna"))
```

> Q. 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 13761 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 4328, 31%
LFC < 0 (down)     : 4474, 33%
outliers [1]       : 0, 0%
low counts [2]     : 0, 0%
(mean count < 0)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

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



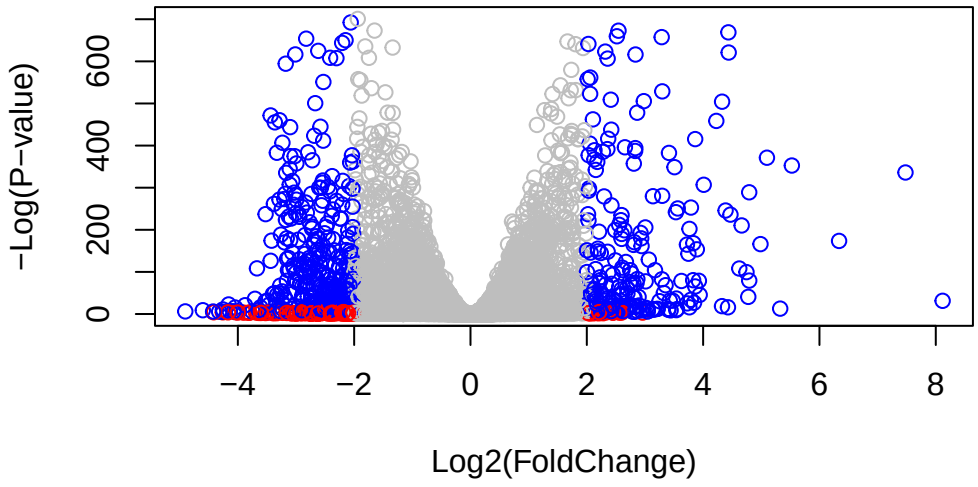
Q. Improve this plot by completing the below code, which adds color and axis labels

```
# Make a color vector for all genes
mycols <- rep("gray", nrow(res) )
```

```
# Color red the genes with absolute fold change above 2
mycols[ abs(res$log2FoldChange) > 2 ] <- "red"

# Color blue those with adjusted p-value less than 0.01
#  and absolute fold change more than 2
inds <- (res$padj < 0.01) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

plot( res$log2FoldChange, -log(res$padj), col= mycols, xlab="Log2(FoldChange)", ylab="-Log(P-
```



Q. 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"
```

```
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   29.9136      0.1802410 0.3128743   0.576081 5.64560e-01
ENSG00000187634  183.2296      0.4259300 0.1357991   3.136471 1.70994e-03
```

```
ENSG00000188976 1651.1881     -0.6927121 0.0549826 -12.598761 2.14486e-36
ENSG00000187961  209.6379      0.7299474 0.1279936   5.702998 1.17718e-08
ENSG00000187583   47.2551      0.0393402 0.2613090   0.150550 8.80330e-01
ENSG00000187642   11.9798      0.5397049 0.5013479   1.076508 2.81700e-01
ENSG00000188290  108.9221      2.0563306 0.1914001  10.743624 6.35019e-27
ENSG00000187608  350.7169      0.2570463 0.1001328   2.567054 1.02567e-02
ENSG00000188157 9128.4394      0.3899096 0.0481440   8.098821 5.54943e-16
ENSG00000131591  156.4791      0.1968739 0.1409590   1.396675 1.62511e-01
                       padj      symbol      entrez                   name
                  <numeric> <character> <character>            <character>
ENSG00000279457 6.53784e-01          NA          NA                     NA
ENSG00000187634 3.52201e-03      SAMD11      148398 sterile alpha motif ..
ENSG00000188976 2.40942e-35       NOC2L       26155 NOC2 like nucleolar ..
ENSG00000187961 4.06810e-08      KLHL17      339451 kelch like family me..
ENSG00000187583 9.12748e-01     PLEKHN1       84069 pleckstrin homology ..
ENSG00000187642 3.68486e-01       PERM1       84808 PPARGC1 and ESRR ind..
ENSG00000188290 5.26099e-26        HES4       57801 hes family bHLH tran..
ENSG00000187608 1.87489e-02       ISG15        9636 ISG15 ubiquitin like..
ENSG00000188157 2.94734e-15        AGRN      375790                  agrin
ENSG00000131591 2.29875e-01    C1orf159       54991 chromosome 1 open re..
```

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

```
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).
##############################################################################
```

```
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"
```

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

```
     1266     54855      1465      2034      2150      6659
-2.422685  3.201862 -2.313714 -1.888000  3.344481  2.392259
```

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

```
attributes(keggres)
```

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

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

```
                                      p.geomean stat.mean        p.val
hsa04110 Cell cycle                 1.888472e-05 -4.205434 1.888472e-05
hsa03030 DNA replication            1.209058e-04 -3.871120 1.209058e-04
hsa04114 Oocyte meiosis             7.921929e-04 -3.206473 7.921929e-04
hsa03440 Homologous recombination   4.227051e-03 -2.734017 4.227051e-03
hsa00010 Glycolysis / Gluconeogenesis 6.053365e-03 -2.563476 6.053365e-03
hsa00240 Pyrimidine metabolism      1.164251e-02 -2.288709 1.164251e-02
                                           q.val set.size         exp1
hsa04110 Cell cycle                 0.002964901      119 1.888472e-05
hsa03030 DNA replication            0.009491108       36 1.209058e-04
hsa04114 Oocyte meiosis             0.041458097       95 7.921929e-04
hsa03440 Homologous recombination   0.165911753       28 4.227051e-03
hsa00010 Glycolysis / Gluconeogenesis 0.190075653       44 6.053365e-03
hsa00240 Pyrimidine metabolism      0.283903993       89 1.164251e-02
```

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

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

```
Info: Working in directory /Users/abrahamrachlin/Documents/BIMM 143/class14
```

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

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

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

```
Warning: reconcile groups sharing member nodes!
```

```
     [,1] [,2]
[1,] "9"  "300"
[2,] "9"  "306"
```

```
Info: Working in directory /Users/abrahamrachlin/Documents/BIMM 143/class14
```

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

```
keggrespathways <- rownames(keggres$greater)[1:5]
```

```
keggresids = substr(keggrespathways, start=1, stop=8)
keggresids
```

```
[1] "hsa04142" "hsa04640" "hsa04630" "hsa04380" "hsa00140"
```

```
pathview(gene.data=foldchanges, pathway.id=keggresids, species="hsa")
```

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

```
Info: Working in directory /Users/abrahamrachlin/Documents/BIMM 143/class14
```

```
Info: Writing image file hsa04142.pathview.png
```

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

Info: Working in directory /Users/abrahamrachlin/Documents/BIMM 143/class14

Info: Writing image file hsa04640.pathview.png

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

Info: Working in directory /Users/abrahamrachlin/Documents/BIMM 143/class14

Info: Writing image file hsa04630.pathview.png

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

Info: Working in directory /Users/abrahamrachlin/Documents/BIMM 143/class14

Info: Writing image file hsa04380.pathview.png

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

Info: Working in directory /Users/abrahamrachlin/Documents/BIMM 143/class14

Info: Writing image file hsa00140.pathview.png
```

Q. Can you do the same procedure as above to plot the pathview figures for the top 5 down-reguled pathways?

```
keggrespathways_down <- rownames(keggres$less)[1:5]
keggresids_down <- substr(keggrespathways_down, start = 1, stop = 8)

print(keggresids_down)
```

```
[1] "hsa04110" "hsa03030" "hsa04114" "hsa03440" "hsa00010"
```

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

```r
gobpsets = go.sets.hs[go.subs.hs$BP]

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

lapply(gobpres, head)
```

```
$greater
                                                      p.geomean stat.mean
GO:0007156 homophilic cell adhesion                   3.574409e-05  4.065745
GO:0016339 calcium-dependent cell-cell adhesion       6.624322e-04  3.414326
GO:0048729 tissue morphogenesis                       9.629642e-04  3.113452
GO:0002009 morphogenesis of an epithelium             1.036665e-03  3.093930
GO:1901617 organic hydroxy compound biosynthetic process 1.825666e-03  2.937016
GO:0035295 tube development                           2.137116e-03  2.867380
                                                             p.val     q.val
GO:0007156 homophilic cell adhesion                   3.574409e-05 0.1348982
GO:0016339 calcium-dependent cell-cell adhesion       6.624322e-04 0.6060058
GO:0048729 tissue morphogenesis                       9.629642e-04 0.6060058
GO:0002009 morphogenesis of an epithelium             1.036665e-03 0.6060058
GO:1901617 organic hydroxy compound biosynthetic process 1.825666e-03 0.6060058
GO:0035295 tube development                           2.137116e-03 0.6060058
                                                      set.size         exp1
GO:0007156 homophilic cell adhesion                         91 3.574409e-05
GO:0016339 calcium-dependent cell-cell adhesion             25 6.624322e-04
GO:0048729 tissue morphogenesis                            356 9.629642e-04
GO:0002009 morphogenesis of an epithelium                  289 1.036665e-03
GO:1901617 organic hydroxy compound biosynthetic process   119 1.825666e-03
GO:0035295 tube development                                335 2.137116e-03

$less
                                          p.geomean stat.mean        p.val
GO:0000279 M phase                     1.070282e-15 -8.081854 1.070282e-15
GO:0048285 organelle fission           1.486831e-14 -7.771854 1.486831e-14
GO:0000280 nuclear division            2.849163e-14 -7.694716 2.849163e-14
GO:0007067 mitosis                     2.849163e-14 -7.694716 2.849163e-14
GO:0000087 M phase of mitotic cell cycle 9.351196e-14 -7.522114 9.351196e-14
GO:0007059 chromosome segregation      2.074373e-11 -6.899759 2.074373e-11
                                              q.val set.size         exp1
GO:0000279 M phase                     4.039243e-12      471 1.070282e-15
GO:0048285 organelle fission           2.688185e-11      362 1.486831e-14
GO:0000280 nuclear division            2.688185e-11      339 2.849163e-14
GO:0007067 mitosis                     2.688185e-11      339 2.849163e-14
```

```
GO:0000087 M phase of mitotic cell cycle 7.058283e-11      349 9.351196e-14
GO:0007059 chromosome segregation        1.304781e-08      136 2.074373e-11

$stats
                                                          stat.mean     exp1
GO:0007156 homophilic cell adhesion                        4.065745 4.065745
GO:0016339 calcium-dependent cell-cell adhesion            3.414326 3.414326
GO:0048729 tissue morphogenesis                            3.113452 3.113452
GO:0002009 morphogenesis of an epithelium                  3.093930 3.093930
GO:1901617 organic hydroxy compound biosynthetic process  2.937016 2.937016
GO:0035295 tube development                                2.867380 2.867380
```

```
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: 8228"
```

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

> 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?

The pathway that is most significant is cell cycle. The most significant does not match, as it is not the same pathway. The factors that could have affected it is potential error in the code on my end.