class 14

Ruofan Kang (A17236920)

```
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


Warning: package 'GenomeInfoDb' was built under R version 4.3.2


Loading required package: SummarizedExperiment


Warning: package 'SummarizedExperiment' was built under R version 4.3.2


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
```

Data Import

Read our counts and metadata CSV files

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

How many genes?

```
nrow(counts)
```

```
[1] 19808
```

```
head(counts, 3)
```

```
                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
                SRR493371
ENSG00000186092         0
ENSG00000279928         0
ENSG00000279457        46
```

Q.How many control and knock-down conditions?

```
table(metadata$condition)
```

```
control_sirna      hoxa1_kd
            3             3
```

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

```
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
```

```
counts <- counts [ , -1]
head (counts, 3)
```

```
                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
```

> 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).

```r
to.rm.inds <- rowSums(counts) == 0
counts <- counts[!to.rm.inds,]
```

> Q. How many genes do we have left?

```r
nrow(counts)
```

```
[1] 15975
```

> 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.

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

~~~
## DESeq setup and analysis

::: {.cell}

```{.r .cell-code}
#1 message:false
#1 warning: false
library(DESeq2)
~~~

:::

```r
#1 Warning:false
dds <- DESeqDataSetFromMatrix(countData= counts,
                              colData= metadata,
                              design= ~condition)
```

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

Run DESeq and get results

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

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

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

```r
dds
```

```
class: DESeqDataSet
dim: 15975 6
metadata(1): version
assays(4): counts mu H cooks
rownames(15975): ENSG00000279457 ENSG00000187634 ... ENSG00000276345
  ENSG00000271254
rowData names(22): baseMean baseVar ... deviance maxCooks
colnames(6): SRR493366 SRR493367 ... SRR493370 SRR493371
colData names(3): id condition sizeFactor
```

> 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.

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

```r
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
```

Quick peak

```r
head(res)
```

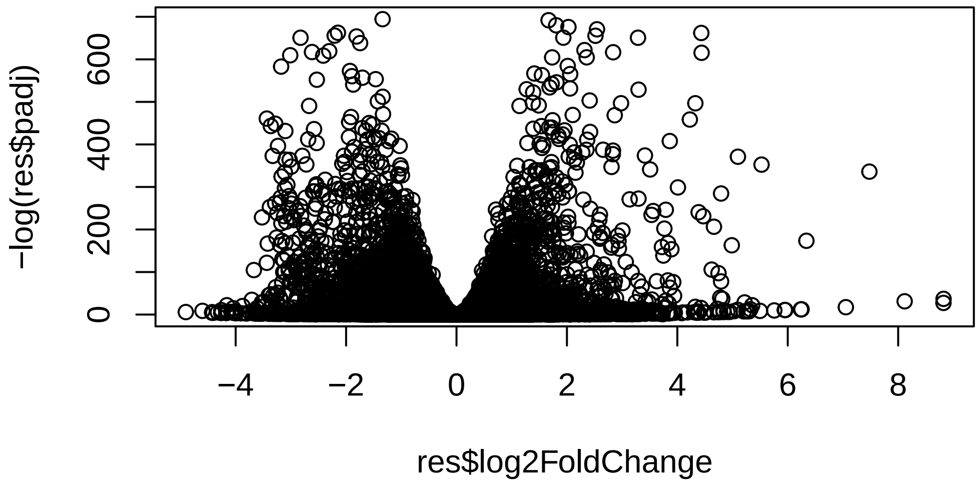
```
log2 fold change (MLE): condition hoxa1_kd vs control_sirna
Wald test p-value: condition hoxa1 kd vs control sirna
DataFrame with 6 rows and 6 columns
                 baseMean log2FoldChange     lfcSE       stat      pvalue
                <numeric>      <numeric> <numeric>  <numeric>   <numeric>
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
```

Add annotation data

```
library(AnnotationDbi)
```

Result visualization

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

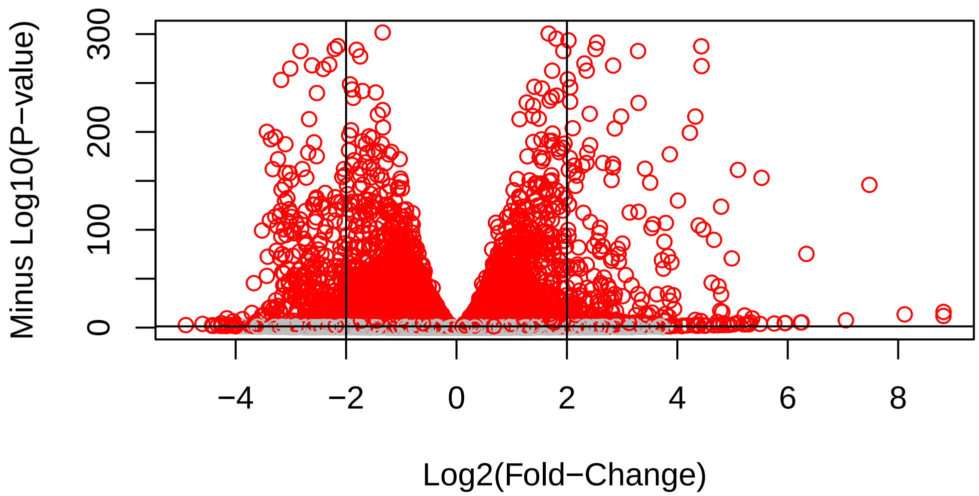


Add some color to this ….

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

```
mycols <- rep("gray", nrow(res))
mycols[res$log2FoldChange > 2] <- "red"
mycols[ res$log2FoldChange< 2] <- "red"
mycols[res$padj > 0.05]<- "gray"
```

```
plot(res$log2FoldChange, -log10(res$padj), col=mycols,
     xlab="Log2(Fold-Change)",
     ylab="Minus Log10(P-value)")
abline(v=c(-2,2))
abline(h=-log10(0.05))
```

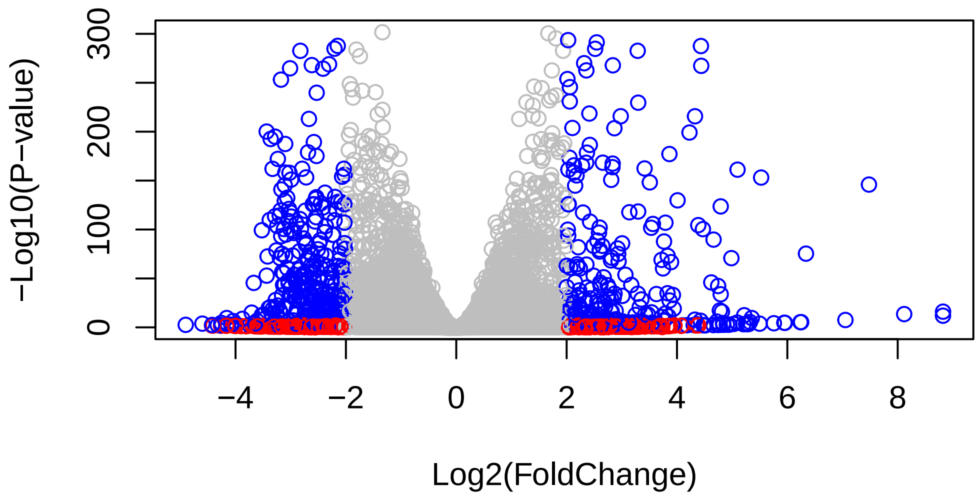


```
# Make a color vector for all genes
mycols <- rep("gray", nrow(res))

# 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
plot(res$log2FoldChange, -log10(res$padj), col=mycols, xlab="Log2(FoldChange)", ylab="-Log
```



Add annotation data

```
library("AnnotationDbi")
```

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

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

```
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
```

```r
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
```

```r
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
```

```r
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 1.43989e-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 0.5215599   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 ..
```

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")
```

```
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
```

```r
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>
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
```

Save results

```r
write.csv(res, file="myresults.rsv")
```

Geneset enchriment

I will use KEGG and GO...

```r
#1 message false
library(gage)
```

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

```r
data(kegg.sets.hs)
```

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

```
     1266     54855      1465     51232      2034      2317
-2.422719  3.201955 -2.313738 -2.059631 -1.888019 -1.649792
```

Run 'gage()' with kegg.sets.hs

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

```r
head(keggres$less, 3)
```

```
                                                  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
                                                      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
                                               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
```

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

```
     1266     54855      1465     51232      2034      2317
-2.422719  3.201955 -2.313738 -2.059631 -1.888019 -1.649792
```

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

```
attributes(keggres)
```

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

```
# Look at the first few down (less) pathways
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                   1.375901e-03 -3.028500
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                   1.375901e-03 0.072234819
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 1.375901e-03
hsa03440 Homologous recombination              28 3.066756e-03
hsa04114 Oocyte meiosis                       102 3.784520e-03
```

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

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

```
Info: Working in directory /Users/nini0029/Desktop/BIMM 143/class 14
```

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

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

```r
# A different PDF based output of the same data
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/nini0029/Desktop/BIMM 143/class 14
```

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

```r
## Focus on top 5 upregulated pathways here for demo purposes only
keggrespathways <- rownames(keggres$greater)[1:5]

# Extract the 8 character long IDs part of each string
keggresids = substr(keggrespathways, start=1, stop=8)
keggresids
```

```
[1] "hsa04060" "hsa05323" "hsa05146" "hsa05332" "hsa04640"
```

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

```
Info: Downloading xml files for hsa04060, 1/1 pathways..

Info: Downloading png files for hsa04060, 1/1 pathways..

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

Info: Working in directory /Users/nini0029/Desktop/BIMM 143/class 14

Info: Writing image file hsa04060.pathview.png

Info: Downloading xml files for hsa05323, 1/1 pathways..

Info: Downloading png files for hsa05323, 1/1 pathways..

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

Info: Working in directory /Users/nini0029/Desktop/BIMM 143/class 14

Info: Writing image file hsa05323.pathview.png

Info: Downloading xml files for hsa05146, 1/1 pathways..

Info: Downloading png files for hsa05146, 1/1 pathways..

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

Info: Working in directory /Users/nini0029/Desktop/BIMM 143/class 14

Info: Writing image file hsa05146.pathview.png

Info: Downloading xml files for hsa05332, 1/1 pathways..

Info: Downloading png files for hsa05332, 1/1 pathways..

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

Info: Working in directory /Users/nini0029/Desktop/BIMM 143/class 14
```

```
Info: Writing image file hsa05332.pathview.png

Info: Downloading xml files for hsa04640, 1/1 pathways..

Info: Downloading png files for hsa04640, 1/1 pathways..

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

Info: Working in directory /Users/nini0029/Desktop/BIMM 143/class 14

Info: Writing image file hsa04640.pathview.png
```

Make my input vector

The top two here (hsa04110, and hsa03030) appear to be the main sets picked out. I will now use 'pathview' to pull these pathways and color up my genes that intersect with these tow pathways

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

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

Info: Working in directory /Users/nini0029/Desktop/BIMM 143/class 14

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

```r
pathview(gene.data=foldchanges, pathway.id = "hsa03030")
```

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

Info: Working in directory /Users/nini0029/Desktop/BIMM 143/class 14

Info: Writing image file hsa03030.pathview.png
```

And insert into my report here:

Go: Gene Ontology

We can do the same style of analysis with Go instead of KEGG here.

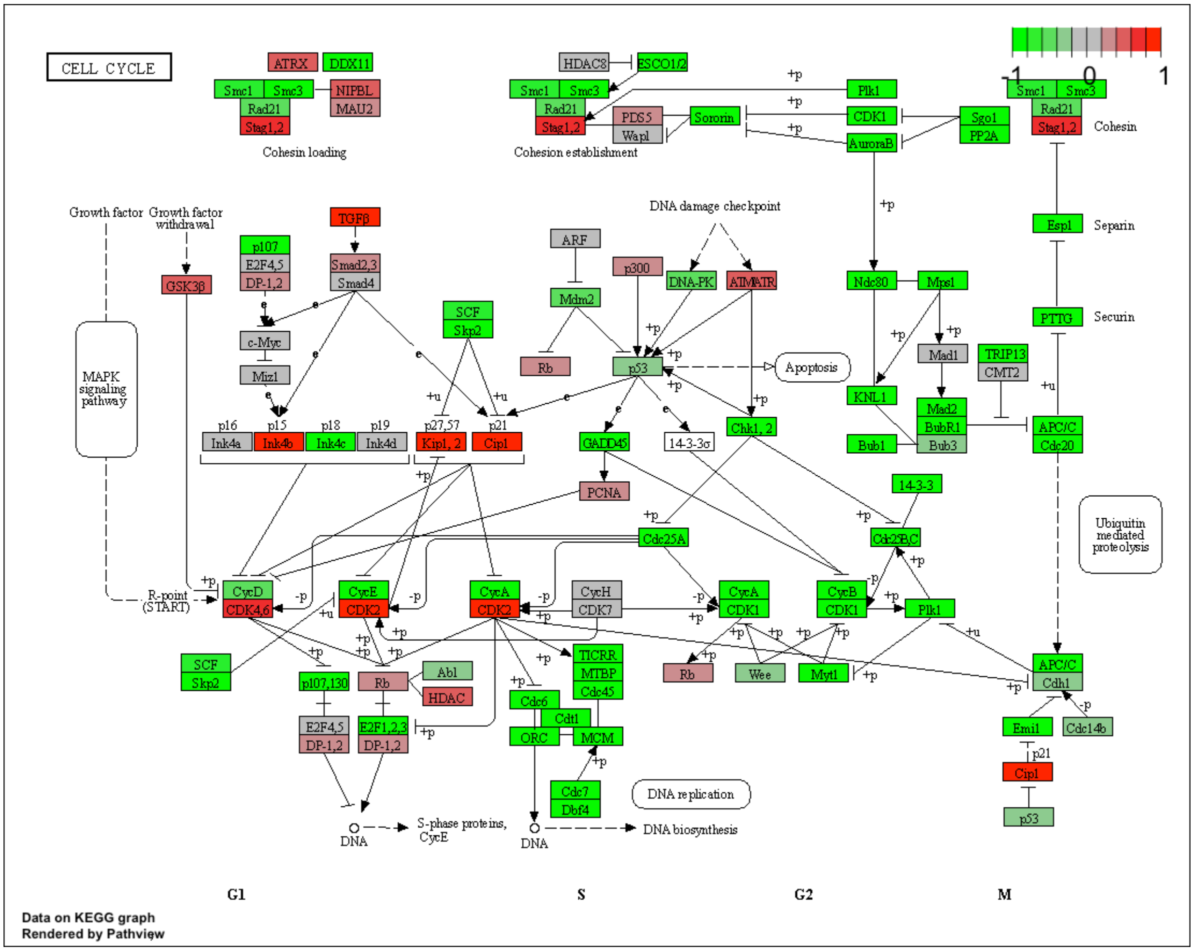


Figure 1: Cell cycle gene

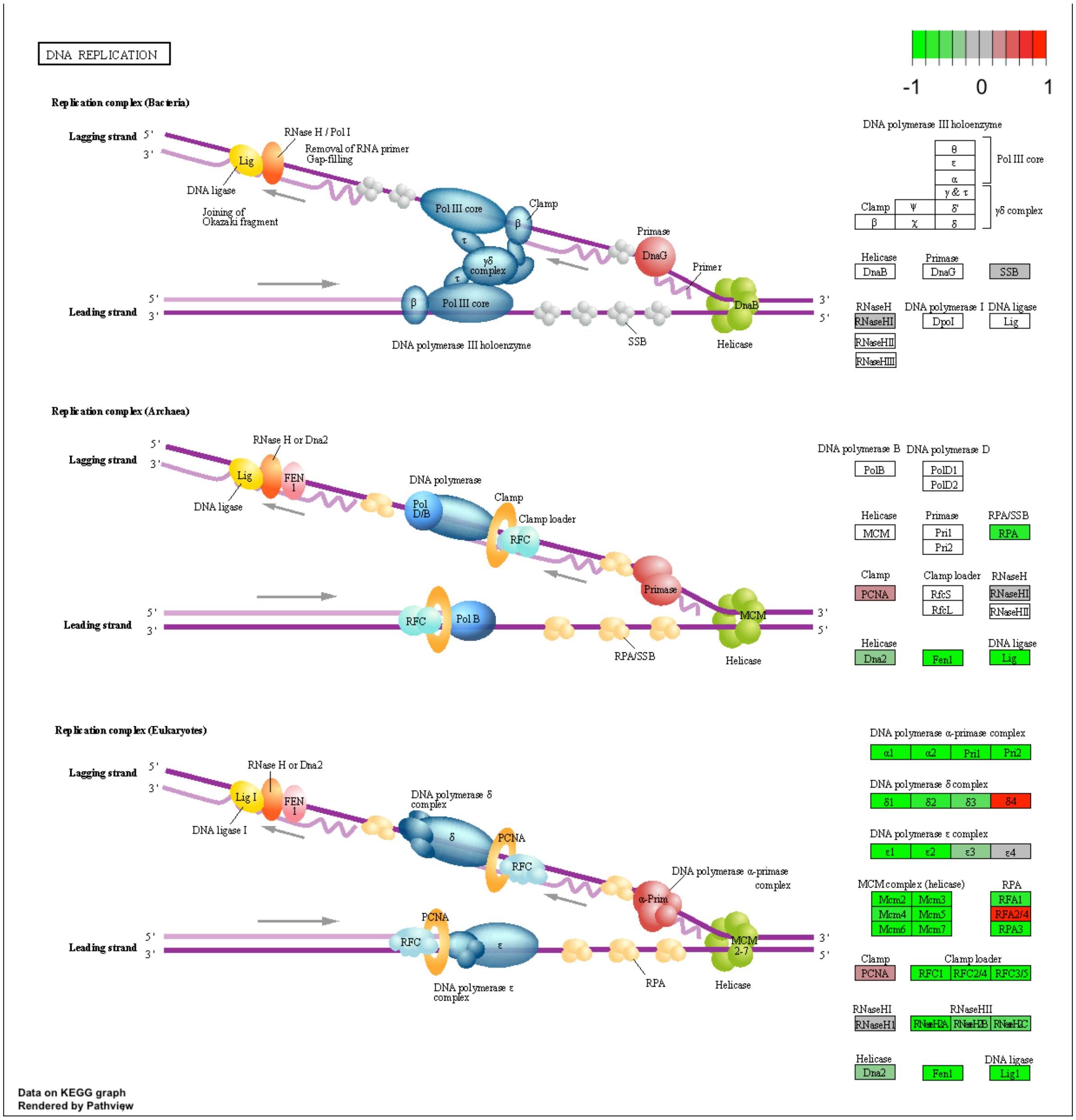


Figure 2: DNA Replication

```
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)
```

Look at our results

```
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             5.843127e-12      376 1.536227e-15
GO:0000280 nuclear division              5.843127e-12      352 4.286961e-15
GO:0007067 mitosis                       5.843127e-12      352 4.286961e-15
GO:0000087 M phase of mitotic cell cycle 1.195965e-11      362 1.169934e-14
GO:0007059 chromosome segregation        1.659009e-08      142 2.028624e-11
GO:0000236 mitotic prometaphase          1.178690e-07       84 1.729553e-10
```

Reactome Analysis

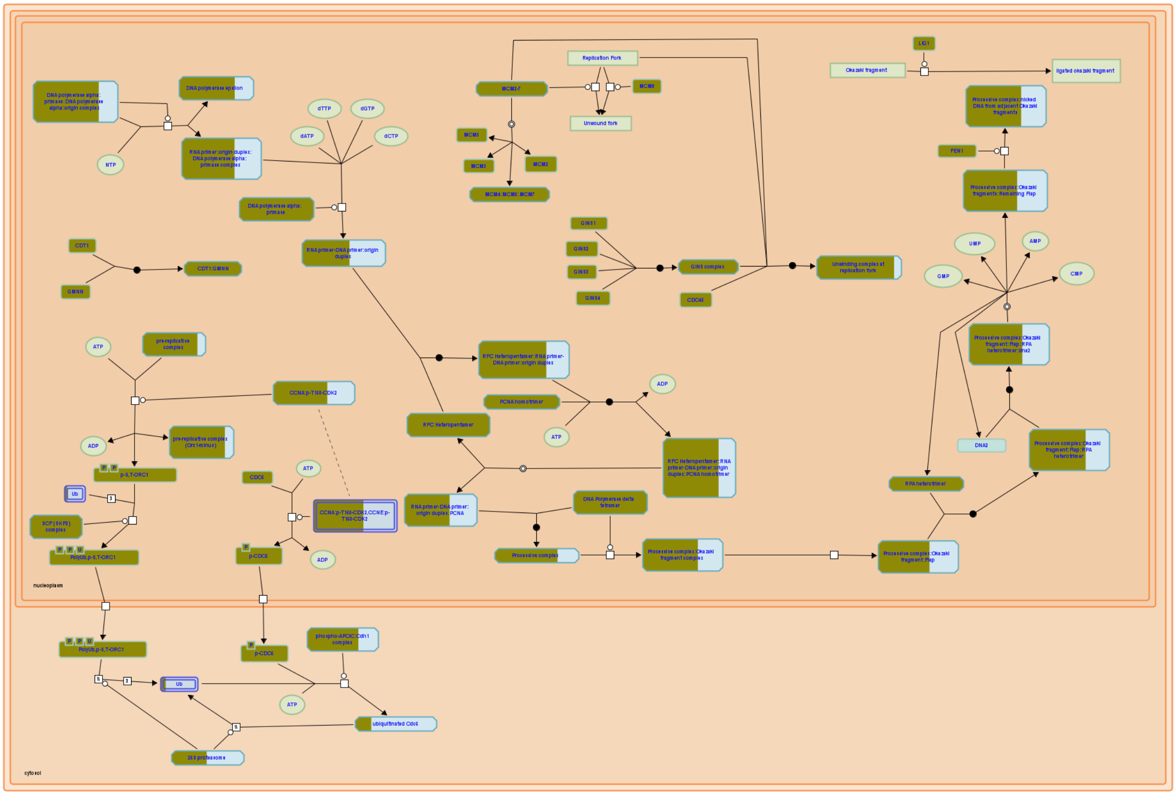
we can use reactome as either its (origional) R package or via it is newer online webserver. The later has some potentially useful pathway viewing functionality so lets try it online. (https://reactome.org/)

To use it online we need a list of significant genes at the alpha <0.05 level as a plain text file.

```
sig_genes <- res[res$padj <= 0.05 & !is.na(res$padj), "symbol"]

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

Now upload this here < https://reactome.org/PathwayBrowser/#TOOL=AT


DNA polymerase alpha: primase:DNA polymerase alpha:origin complex
DNA polymerase epsilon
RNA primer:origin duplex: DNA polymerase alpha: primase complex
NTP
dTTP
dGTP
dATP
dCTP
DNA polymerase alpha: primase
RNA primer-DNA primer:origin duplex
CDT1
GMNN
CDT1:GMNN
Replication Fork
MCM2-7
MCM8
Unwound fork
MCM5
MCM3
MCM2
MCM4:MCM6:MCM7
GINS1
GINS2
GINS3
GINS4
GINS complex
CDC45
Unwinding complex at replication fork
LIG1
Okazaki fragment
ligated okazaki fragment
Processive complex:nicked DNA from adjacent Okazaki fragments
FEN1
Processive complex:Okazaki fragments:Remaining Flap
UMP
AMP
GMP
CMP
Processive complex:Okazaki fragment:Flap:RPA heterotrimer:dna2
DNA2
Processive complex:Okazaki fragment:Flap:RPA heterotrimer
RPA heterotrimer
Processive complex:Okazaki fragment:Flap
ATP
pre-replicative complex
CCNA:p-T160-CDK2
ADP
pre-replicative complex (Orc1-minus)
P
P
p-S,T-ORC1
Ub
3
SCF(SKP2) complex
P
P
U
PolyUb,p-S,T-ORC1
CDC6
ATP
CCNA:p-T160-CDK2,CCNE:p-T160-CDK2
P
p-CDC6
ADP
RFC Heteropentamer:RNA primer-DNA primer:origin duplex
PCNA homotrimer
ADP
ATP
RFC Heteropentamer
RFC Heteropentamer:RNA primer-DNA primer:origin duplex:PCNA homotrimer
RNA primer-DNA primer: origin duplex:PCNA
DNA Polymerase delta tetramer
Processive complex
Processive complex:Okazaki fragment complex
nucleoplasm
P
P
U
PolyUb,p-S,T-ORC1
P
p-CDC6
phospho-APC/C:Cdh1 complex
3
Ub
ATP
ubiquitinated Cdc6
26S proteasome
cytosol

Synthesis of DNA: