DESeq2 analysis mini project

Soobin (PID:A15201229)

2/24/2022

Here we will work on a complete differential expression analysis project. We will use DESeq2 for this.

```
library(DESeq2)
library(ggplot2)
library(AnnotationDbi)
library(org.Hs.eg.db)
```

1. Input the counts and metadata files

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

Inspect these objects.

```
head(colData)
```

```
##          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(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 <- 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. Check on correspondence of colData and countData

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

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

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).

```
counts <- countData[ rowSums(countData) != 0, ]
head(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
```

Run DESeq2 analysis

The steps here are to first setup the object required by DESeq using the `DESeqDataSetFromMatrix()` function. Then, I can run my differential expression with `DESeq()`.

```
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)
```

Now get my results out of this dds object.

```
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
```

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

Add annotation

```
columns(org.Hs.eg.db)
```

```
##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
##  [6] "ENTREZID"     "ENZYME"       "EVIDENCE"     "EVIDENCEALL"  "GENENAME"
## [11] "GO"           "GOALL"        "IPI"          "MAP"          "OMIM"
## [16] "ONTOLOGY"     "ONTOLOGYALL"  "PATH"         "PFAM"         "PMID"
## [21] "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"       "UNIGENE"
## [26] "UNIPROT"
```

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
```

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

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

Check my result.

```
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      0.1792570 0.3248225   0.551861 5.81043e-01
## ENSG00000187634  183.2296      0.4264571 0.1402660   3.040345 2.36307e-03
## ENSG00000188976 1651.1881     -0.6927205 0.0548462 -12.630233 1.43852e-36
## ENSG00000187961  209.6379      0.7297556 0.1318601   5.534318 3.12441e-08
## ENSG00000187583   47.2551      0.0405766 0.2718936   0.149237 8.81367e-01
## ENSG00000187642   11.9798      0.5428107 0.5215615   1.040742 2.97995e-01
##                       padj      symbol      entrez              genename
##                  <numeric> <character> <character>           <character>
## ENSG00000279457 6.86556e-01      WASH9P   102723897 WAS protein family h..
## ENSG00000187634 5.15726e-03      SAMD11      148398 sterile alpha motif ..
## ENSG00000188976 1.76381e-35       NOC2L       26155 NOC2 like nucleolar ..
## ENSG00000187961 1.13418e-07      KLHL17      339451 kelch like family me..
## ENSG00000187583 9.19031e-01     PLEKHN1       84069 pleckstrin homology ..
## ENSG00000187642 4.03380e-01       PERM1       84808 PPARGC1 and ESRR ind..
```

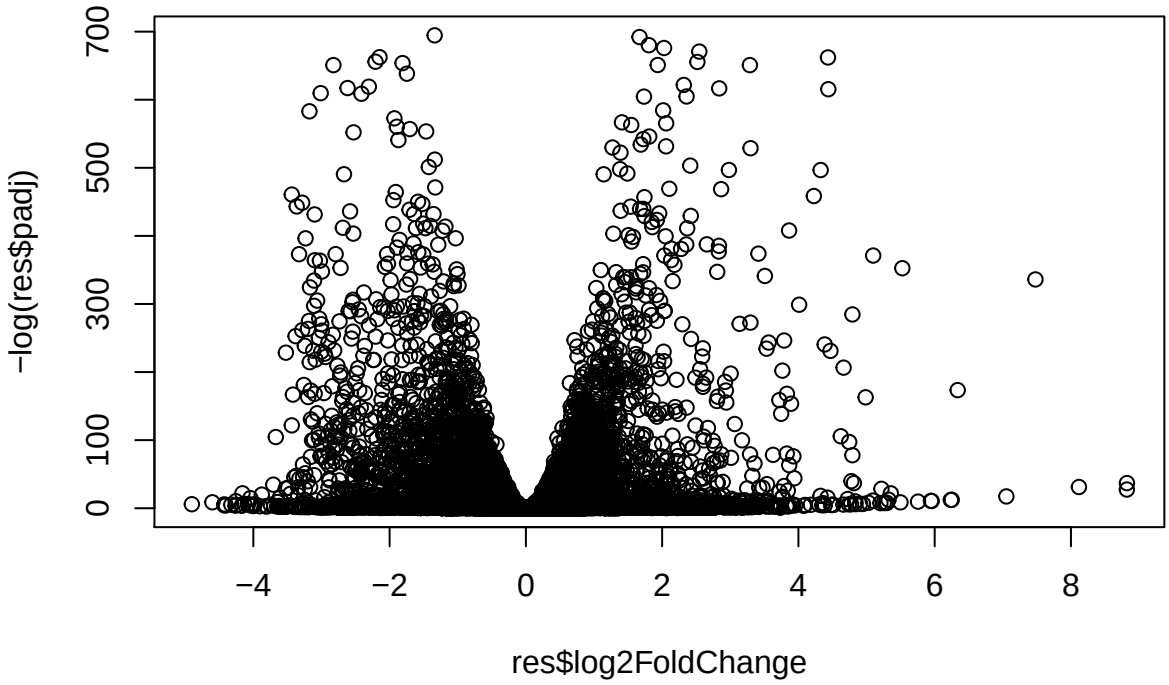
> Q. Finally for this section let's reorder these results by adjusted p-value and save them to a CSV file in your current project directory.

```
# res = res[order(res$pvalue),]
# write.csv(res, file="deseq_results3.csv")
```

Volcano plot

Common summary figure that gives a nice overview of our result.

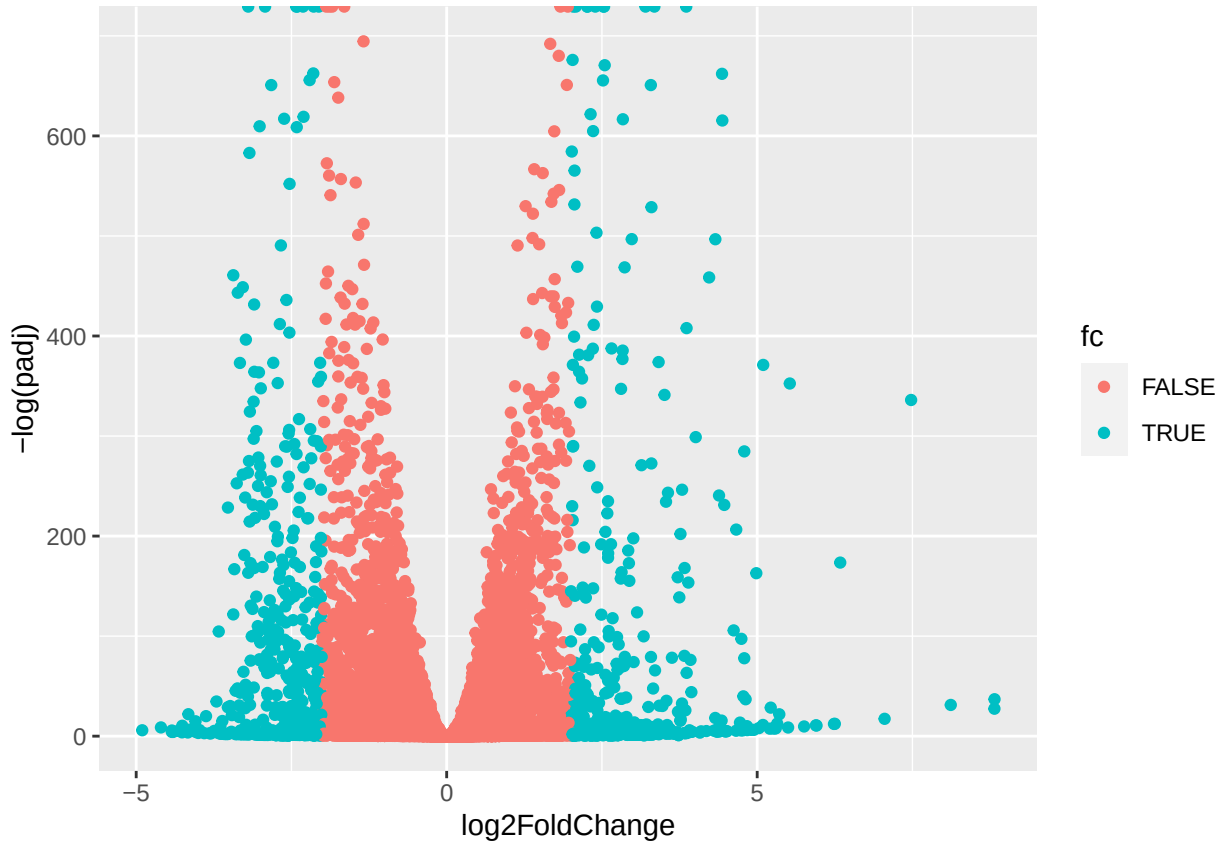
```
plot(res$log2FoldChange, -log(res$padj))
```



Try ggplot for this.

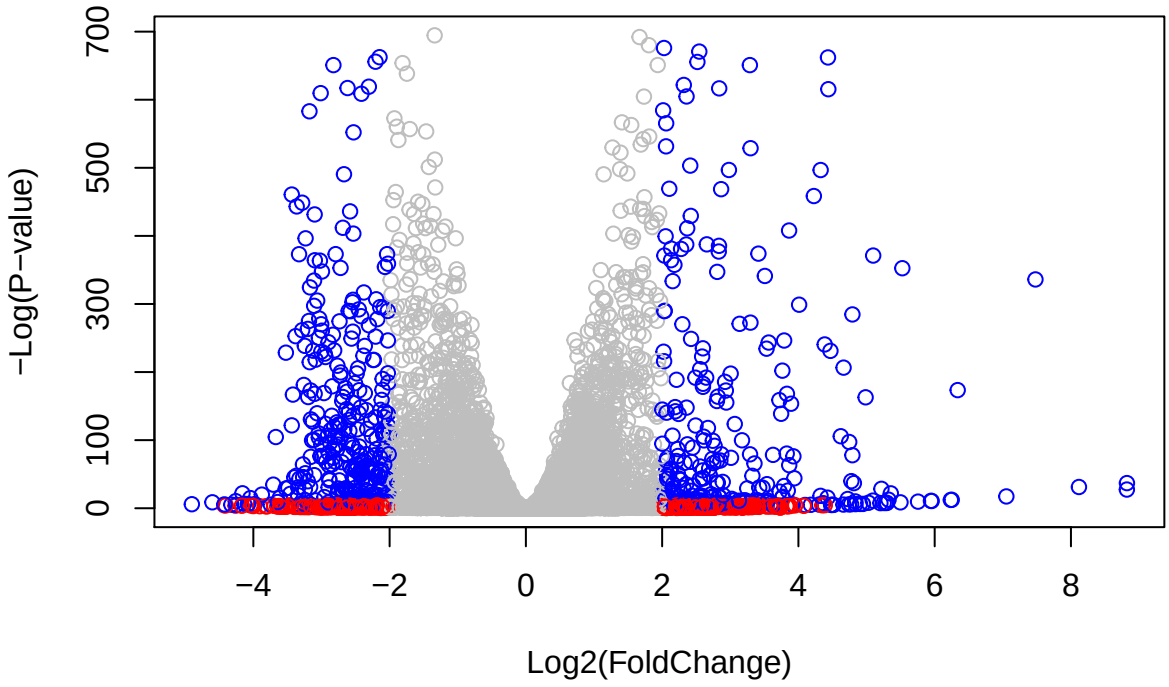
```
tmp <- as.data.frame(res)
tmp$fc <- abs(res$log2FoldChange) > 2

ggplot(tmp) + aes(x=log2FoldChange, y= -log(padj), col=fc) + geom_point()
```

```
## Warning: Removed 1237 rows containing missing values (geom_point).
```



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-value)" )
```




```
# library(EnhancedVolcano)

# x <- as.data.frame(res)

# EnhancedVolcano(x,lab = x$symbol,x = 'log2FoldChange',y = 'pvalue')
```

Pathway analysis and gene set enrichment

Here we try to bring back the biology and help with the interpretation of our results. We try to answer the question: which pathways and functions feature heavily in our differentially expressed genes.

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

Recall that we need a "vector of importance" as input for GAGE that has ENTREZ ids set as a name of attribute.

```
foldchange <- res$log2FoldChange
names(foldchange) <- res$entrez
```

```
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"
```

Now, let's run the gage pathway analysis.

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

Check the attributes of keggres.

```
attributes(keggres)
```

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

Look at the first 2 down-regulated pathways.

```
# Look at the first few down(less) pathways.
head(keggres$less, 2)
```

```
##                             p.geomean stat.mean        p.val       q.val
## hsa04110 Cell cycle      8.995726e-06 -4.378644 8.995726e-06 0.001448312
## hsa03030 DNA replication 9.424075e-05 -3.951803 9.424075e-05 0.007586380
##                          set.size         exp1
## hsa04110 Cell cycle           121 8.995726e-06
## hsa03030 DNA replication       36 9.424075e-05
```

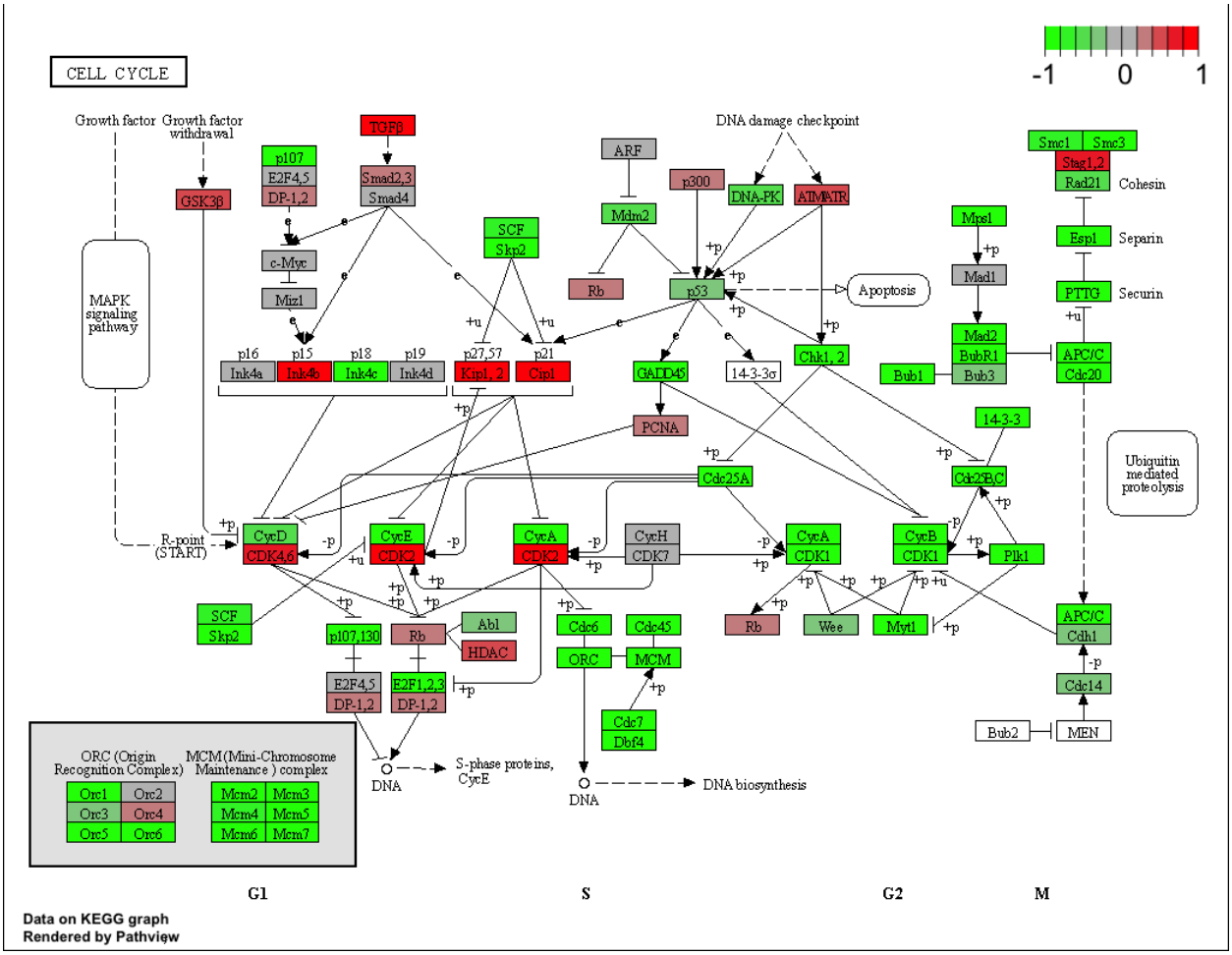
Now, let's try out the `pathview()` function from the pathview package to make a pathway plot with our RNA-Seq expression results shown in color.

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

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

```
## Info: Working in directory /Users/sbhwang/Desktop/BIMM 143/DESeq2 analysis mini-project
```

```
## 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-regulated pathways?

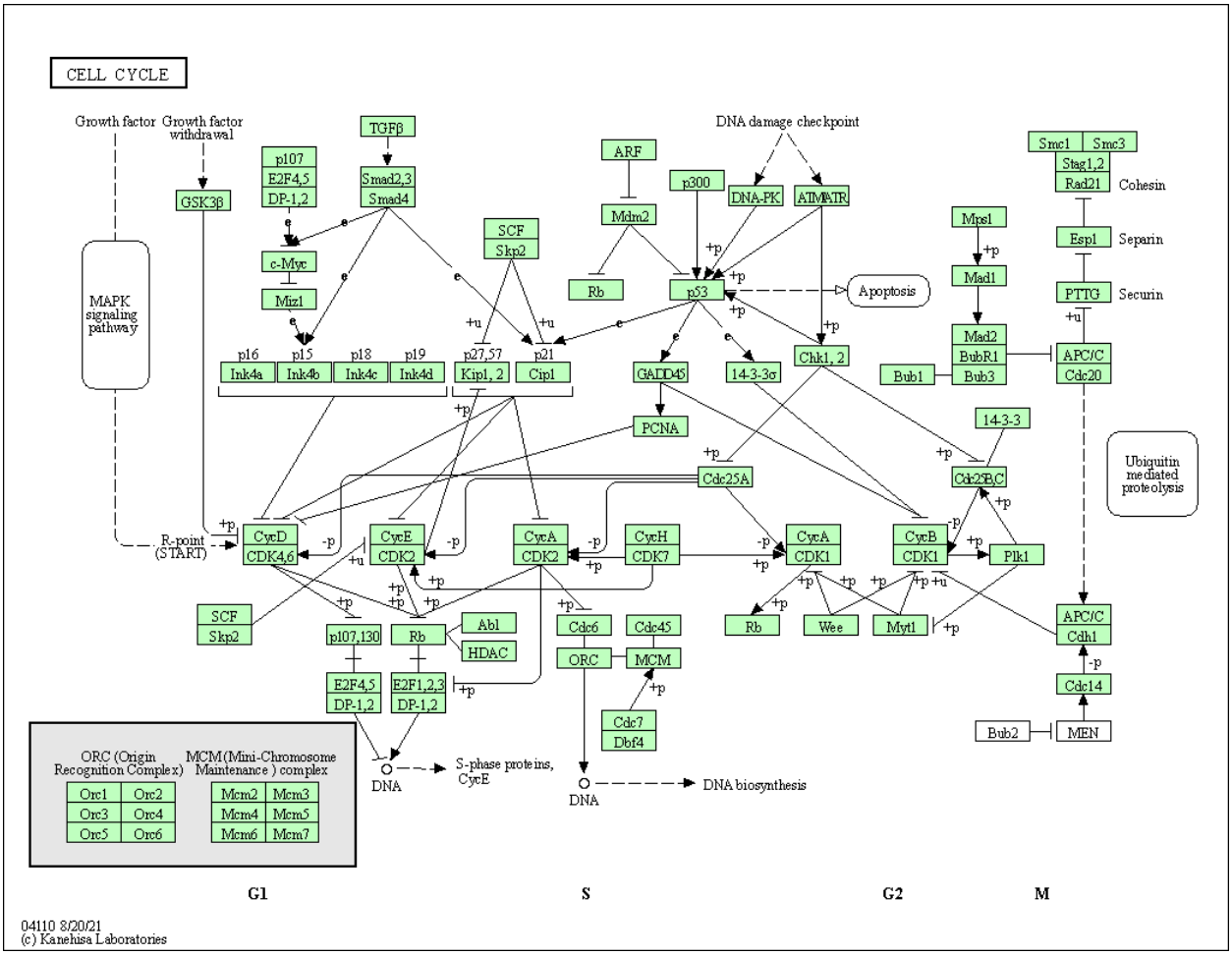
```
# 5 down-regulated pathways
keggrespathways <- rownames(keggres$less)[1:5]

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

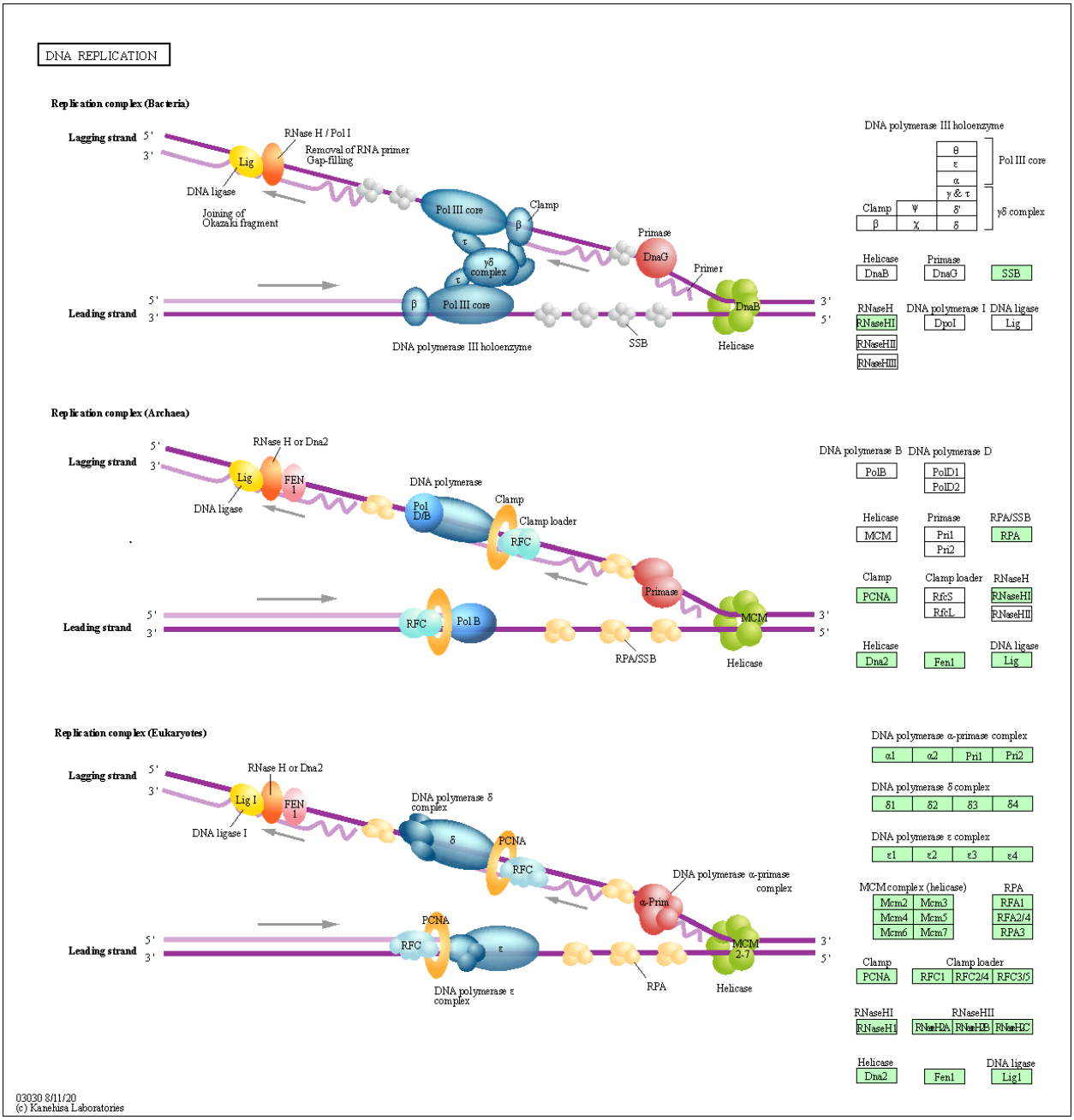
```
## [1] "hsa04110" "hsa03030" "hsa03013" "hsa03440" "hsa04114"
```

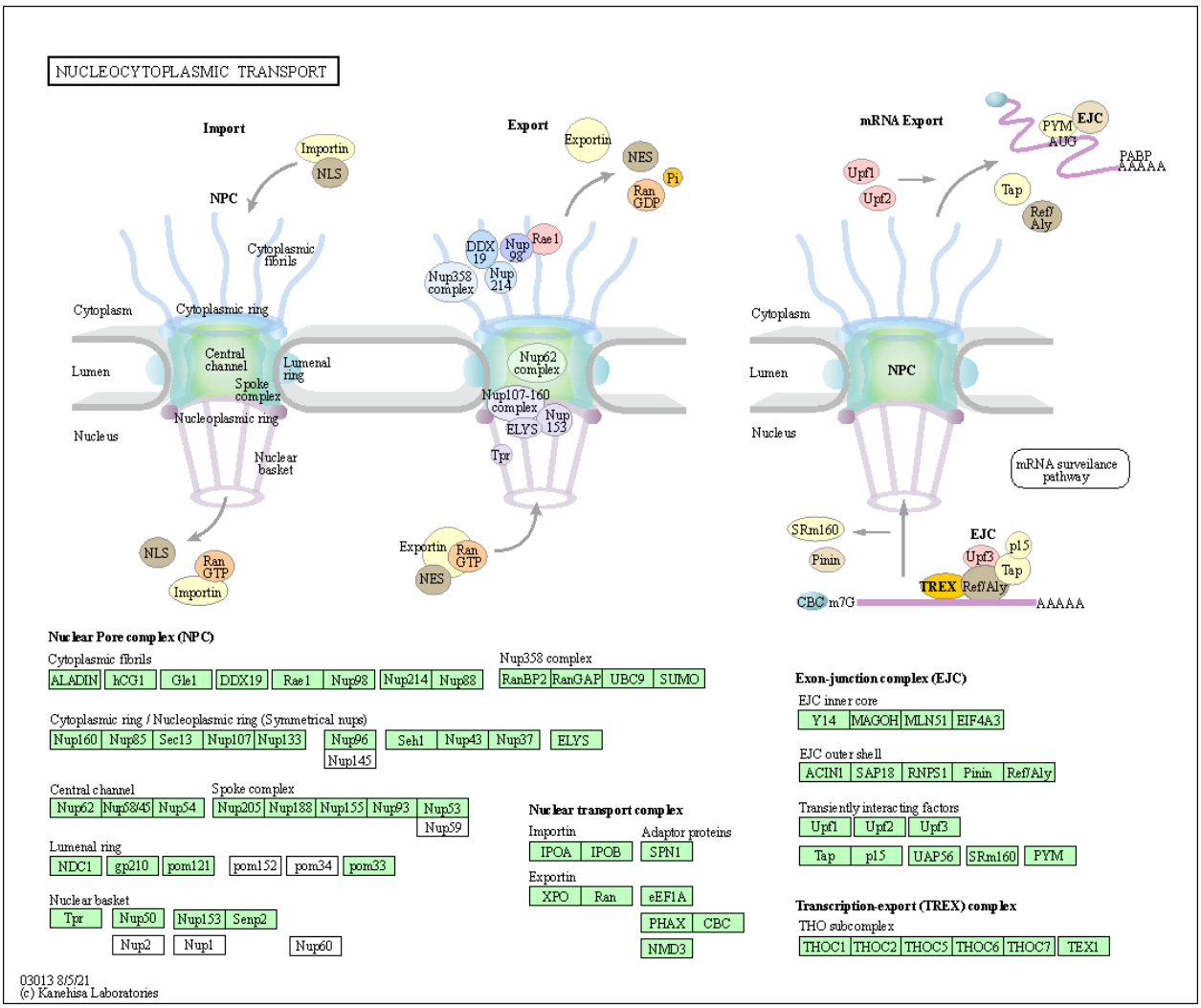
```
# View the pathway by passing 5 IDs in keggresids
pathview(gene.data = foldchange, pathway.id = keggresids, species = "hsa")
```

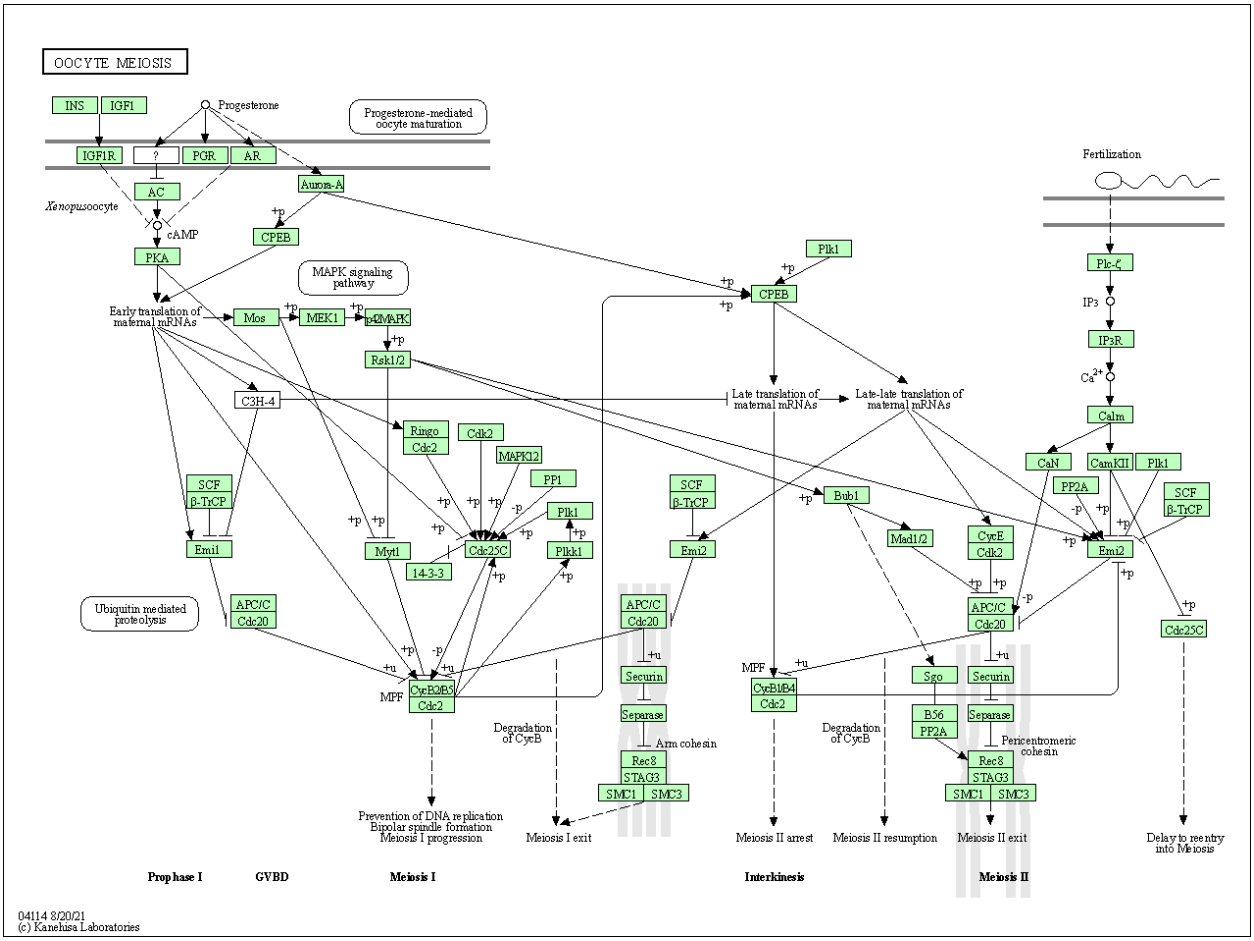
```
## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/sbhwang/Desktop/BIMM 143/DESeq2 analysis mini-project

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

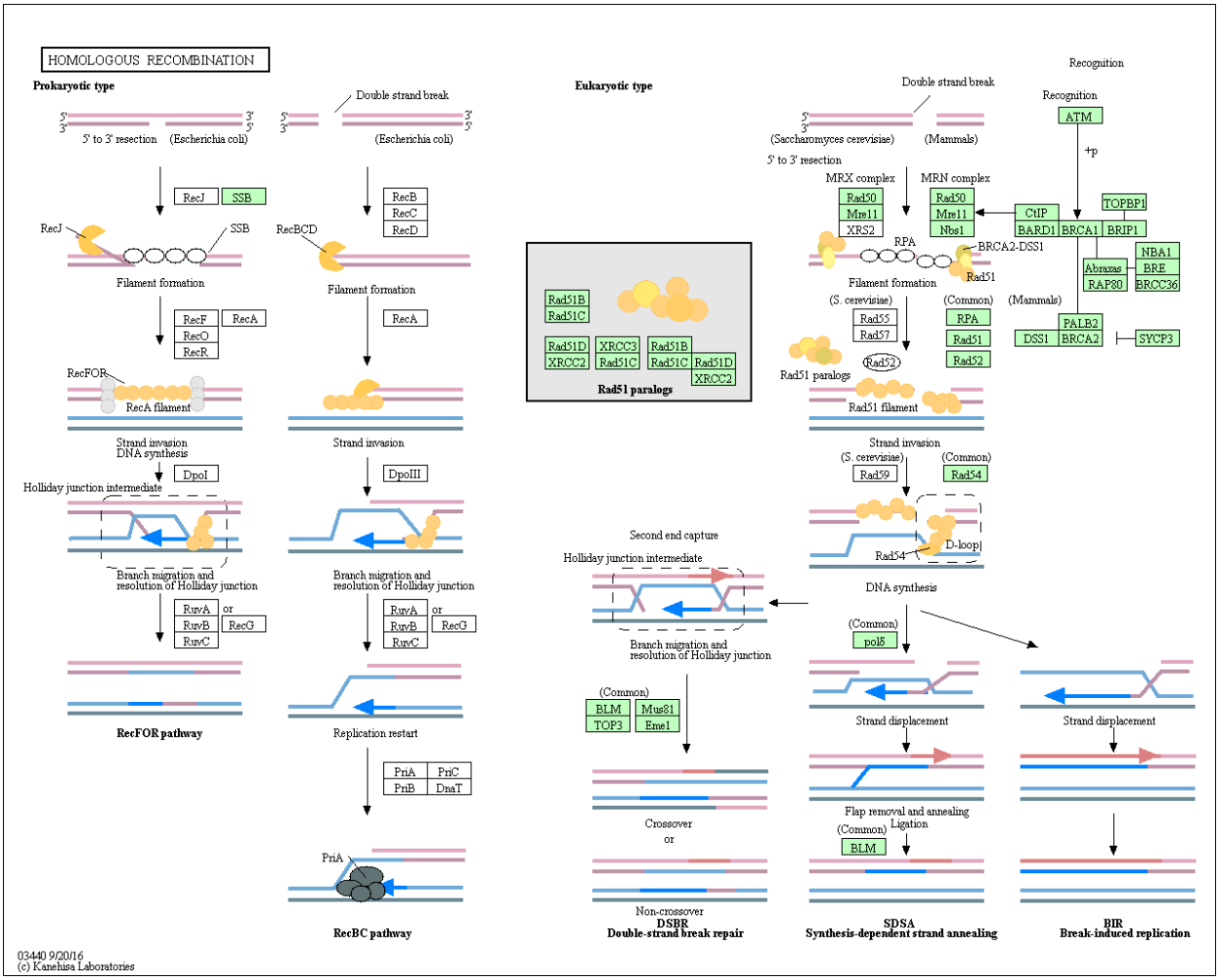
```
## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/sbhwang/Desktop/BIMM 143/DESeq2 analysis mini-project

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

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

## Info: Working in directory /Users/sbhwang/Desktop/BIMM 143/DESeq2 analysis mini-project

## Info: Writing image file hsa03013.pathview.png

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

## Info: Working in directory /Users/sbhwang/Desktop/BIMM 143/DESeq2 analysis mini-project

## Info: Writing image file hsa03440.pathview.png

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

## Info: Working in directory /Users/sbhwang/Desktop/BIMM 143/DESeq2 analysis mini-project

## Info: Writing image file hsa04114.pathview.png
```




DNA REPLICATION
Replication complex (Bacteria)
Lagging strand
Leading strand
RNase H / Pol I
Removal of RNA primer
Gap-filling
Lig
DNA ligase
Joining of
Okazaki fragment
Pol III core
Clamp
β
τ
γδ
complex
Primase
DnaG
Primer
DnaB
SSB
Helicase
DNA polymerase III holoenzyme
DNA polymerase III holoenzyme
θ
ε
α
γ & τ
Pol III core
Clamp
ψ
δ'
γδ complex
β
χ
δ
Helicase
DnaB
Primase
DnaG
SSB
RNaseH
RNaseHII
RNaseHII
RNaseHIII
DNA polymerase I
DpoI
DNA ligase
Lig
Replication complex (Archaea)
RNase H or Dna2
Lig
FEN
1
DNA ligase
DNA polymerase
Pol
D/B
Clamp
Clamp loader
RFC
Primase
MCM
RFC
Pol B
RPA/SSB
Helicase
DNA polymerase B
PolB
DNA polymerase D
PolD1
PolD2
Helicase
MCM
Primase
Pri1
Pri2
RPA/SSB
RPA
Clamp
PCNA
Clamp loader
RfcS
RfcL
RNaseH
RNaseHI
RNaseHII
Helicase
Dna2
Fen1
DNA ligase
Lig
Replication complex (Eukaryotes)
RNase H or Dna2
Lig I
FEN
1
DNA ligase I
DNA polymerase δ
complex
δ
PCNA
RFC
DNA polymerase α-primase
complex
α-Prim
MCM
2-7
RFC
PCNA
ε
DNA polymerase ε
complex
RPA
Helicase
DNA polymerase α-primase complex
α1
α2
Pri1
Pri2
DNA polymerase δ complex
δ1
δ2
δ3
δ4
DNA polymerase ε complex
ε1
ε2
ε3
ε4
MCM complex (helicase)
Mcm2
Mcm3
Mcm4
Mcm5
Mcm6
Mcm7
RPA
RFA1
RFA2/4
RPA3
Clamp
PCNA
Clamp loader
RFC1
RFC2/4
RFC3/5
RNaseHI
RNaseHI
RNaseHII
RNaseHIIA
RNaseHIIB
RNaseHIIC
Helicase
Dna2
Fen1
DNA ligase
Lig1
03030 8/11/20
(c) Kanehisa Laboratories


NUCLEOCYTOPLASMIC TRANSPORT
Import
Importin
NLS
NPC
Cytoplasmic fibrils
Cytoplasm
Cytoplasmic ring
Central channel
Spoke complex
Lumenal ring
Lumen
Nucleoplasmic ring
Nucleus
Nuclear basket
NLS
Ran GTP
Importin
Export
Exportin
NES
Pi
Ran GDP
DDX 19
Nup 98
Rae1
Nup358 complex
Nup 214
Nup62 complex
Nup107-160 complex
ELYS
Nup 153
Tpr
Exportin
Ran GTP
NES
mRNA Export
EJC
PYM
AUG
PABP
AAAAA
Upf1
Upf2
Tap
Ref/ Aly
Cytoplasm
Lumen
NPC
Nucleus
mRNA surveilance pathway
SRm160
Pinin
EJC
Upf3
p15
Tap
TREX
Ref/Aly
CBC
m7G
AAAAA
Nuclear Pore complex (NPC)
Cytoplasmic fibrils
ALADIN
hCG1
Gle1
DDX19
Rae1
Nup98
Nup214
Nup88
Nup358 complex
RanBP2
RanGAP
UBC9
SUMO
Cytoplasmic ring / Nucleoplasmic ring (Symmetrical nups)
Nup160
Nup85
Sec13
Nup107
Nup133
Nup96
Nup145
Seh1
Nup43
Nup37
ELYS
Central channel
Nup62
Nup58/45
Nup54
Spoke complex
Nup205
Nup188
Nup155
Nup93
Nup53
Nup59
Lumenal ring
NDC1
gp210
pom121
pom152
pom34
pom33
Nuclear basket
Tpr
Nup50
Nup153
Senp2
Nup2
Nup1
Nup60
Nuclear transport complex
Importin
Adaptor proteins
IPOA
IPOB
SPN1
Exportin
XPO
Ran
eEF1A
PHAX
CBC
NMD3
Exon-junction complex (EJC)
EJC inner core
Y14
MAGOH
MLN51
EIF4A3
EJC outer shell
ACIN1
SAP18
RNPS1
Pinin
Ref/Aly
Transiently interacting factors
Upf1
Upf2
Upf3
Tap
p15
UAP56
SRm160
PYM
Transcription-export (TREX) complex
THO subcomplex
THOC1
THOC2
THOC5
THOC6
THOC7
TEX1
03013 8/5/21
(c) Kanehisa Laboratories


OOCYTE MEIOSIS
INS
IGF1
Progesterone
Progesterone-mediated oocyte maturation
IGF1R
?
PGR
AR
AC
Xenopus oocyte
cAMP
PKA
Aurora-A
CPEB
MAPK signaling pathway
Early translation of maternal mRNAs
Mos
MEK1
p42MAPK
Rsk1/2
C3H-4
Ringo
Cdc2
Cdk2
MAPK12
PP1
Plk1
SCF
β-TrCP
Emi1
Myt1
14-3-3
Cdc25C
Plkk1
Ubiquitin mediated proteolysis
APC/C
Cdc20
MPF
CycB2/B5
Cdc2
Degradation of CycB
Securin
Separase
Arm cohesin
Rec8
STAG3
SMC1
SMC3
Prevention of DNA replication
Bipolar spindle formation
Meiosis I progression
Meiosis I exit
Late translation of maternal mRNAs
Late-late translation of maternal mRNAs
Bub1
Emi2
CycB1/B4
Cdc2
Meiosis II arrest
Mad1/2
CycE
Cdk2
Sgo
B56
PP2A
Pericentromeric cohesin
Meiosis II resumption
Meiosis II exit
Fertilization
Plc-ζ
IP3
IP3R
Ca2+
Calm
CaN
CamKII
Delay to reentry into Meiosis
Prophase I
GVBD
Meiosis I
Interkinesis
Meiosis II
04114 8/20/21
(c) Kanehisa Laboratories


HOMOLOGOUS RECOMBINATION
Prokaryotic type
Double strand break
5' to 3' resection (Escherichia coli)
(Escherichia coli)
RecJ
SSB
RecB
RecC
RecD
RecBCD
Filament formation
RecF
RecO
RecR
RecA
RecFOR
RecA filament
Strand invasion
DNA synthesis
DpoI
DpoIII
Holliday junction intermediate
Branch migration and resolution of Holliday junction
RuvA
RuvB
RuvC
or
RecG
RecFOR pathway
Replication restart
PriA
PriB
PriC
DnaT
RecBC pathway
Eukaryotic type
Rad51B
Rad51C
XRCC3
Rad51D
XRCC2
Rad51 paralogs
Recognition
ATM
+p
(Saccharomyces cerevisiae)
(Mammals)
5' to 3' resection
MRX complex
MRN complex
Rad50
Mre11
XRS2
Nbs1
CtIP
BARD1
BRCA1
BRIP1
TOPBP1
RPA
BRCA2-DSS1
Rad51
NBA1
BRE
BRCC36
Abraxas
RAP80
PALB2
DSS1
BRCA2
SYCP3
(S. cerevisiae)
(Common)
Rad55
Rad57
Rad52
Rad51 paralogs
Rad51 filament
Rad59
Rad54
D-loop
DNA synthesis
pol δ
Second end capture
Crossover
Non-crossover
BLM
TOP3
Mus81
Eme1
Strand displacement
Flap removal and annealing
Ligation
DSBR
Double-strand break repair
SDSA
Synthesis-dependent strand annealing
BIR
Break-induced replication
03440 9/20/16
(c) Kanehisa Laboratories