

# Hw 4

## 1. Load Data

```
data <- read.csv('GSE37704.csv')
rownames(data) <- data[,1]
data <- data[,-1]
```

## 2. Preprocess Data

### 2.1 Normalize Data by count per million

```
normData <- lapply(data, function(x){return(x/sum(x) * 1000000)})
df <- as.data.frame(normData, row.names = rownames(data), col.names = names(normData))
summary.data.frame(df)
```

```
##      control_1      control_2      control_3
## Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
## 1st Qu.: 0.070   1st Qu.: 0.123   1st Qu.: 0.098
## Median : 6.603   Median : 6.471   Median : 6.390
## Mean   : 50.485   Mean   : 50.485   Mean   : 50.485
## 3rd Qu.: 36.315   3rd Qu.: 36.545   3rd Qu.: 36.194
## Max.   :28286.585   Max.   :27042.737   Max.   :27266.460
##      Hoxa1KN_1      Hoxa1KN_2      Hoxa1KN_3
## Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
## 1st Qu.: 0.119   1st Qu.: 0.106   1st Qu.: 0.092
## Median : 6.084   Median : 6.119   Median : 6.075
## Mean   : 50.485   Mean   : 50.485   Mean   : 50.485
## 3rd Qu.: 35.967   3rd Qu.: 35.702   3rd Qu.: 36.268
## Max.   :25486.423   Max.   :26528.477   Max.   :24289.459
```

### 2.2 Log normalized data

```
logDf <- log(df+1)
summary.data.frame(logDf)
```

```
##      control_1      control_2      control_3      Hoxa1KN_1
## Min.   : 0.00000   Min.   : 0.0000   Min.   : 0.00000   Min.   : 0.0000
## 1st Qu.: 0.06789   1st Qu.: 0.1162   1st Qu.: 0.09309   1st Qu.: 0.1127
## Median : 2.02851   Median : 2.0110   Median : 2.00013   Median : 1.9578
## Mean   : 2.07267   Mean   : 2.0759   Mean   : 2.07095   Mean   : 2.0668
## 3rd Qu.: 3.61940   3rd Qu.: 3.6255   3rd Qu.: 3.61614   3rd Qu.: 3.6100
## Max.   :10.25018   Max.   :10.2052   Max.   :10.21345   Max.   :10.1459
##      Hoxa1KN_2      Hoxa1KN_3
## Min.   : 0.0000   Min.   : 0.00000
## 1st Qu.: 0.1011   1st Qu.: 0.08806
## Median : 1.9627   Median : 1.95661
## Mean   : 2.0686   Mean   : 2.07094
```

```
## 3rd Qu.: 3.6028 3rd Qu.: 3.61813
## Max. :10.1860 Max. :10.09784
```

## 2.3 cut data

```
s <- df %>%
  dplyr::add_rownames() %>%
  dplyr::filter( (control_1 >= 5) | (control_2 >= 5) | (control_3 >= 5)
    | (Hoxa1KN_1 >= 5) | (Hoxa1KN_2 >= 5) | (Hoxa1KN_3 >= 5)) %>%
  dplyr::select(rowname)
```

```
## Warning: Deprecated, use tibble::rownames_to_column() instead.
```

```
cuttedData <- data %>%
  dplyr::add_rownames() %>%
  dplyr::inner_join(s, by='rowname')
```

```
## Warning: Deprecated, use tibble::rownames_to_column() instead.
```

```
cuttedData <- as.data.frame(cuttedData)
rownames(cuttedData) <- cuttedData[,1]
cuttedData <- cuttedData[,-1]
print("")
```

```
## [1] ""
```

```
nrow(cuttedData)
```

```
## [1] 10859
```

```
summary(cuttedData)
```

```
## control_1 control_2 control_3 Hoxa1KN_1
## Min. : 0 Min. : 0 Min. : 0 Min. : 0
## 1st Qu.: 196 1st Qu.: 225 1st Qu.: 282 1st Qu.: 218
## Median : 445 Median : 510 Median : 642 Median : 520
## Mean : 1305 Mean : 1488 Mean : 1880 Mean : 1537
## 3rd Qu.: 1018 3rd Qu.: 1172 3rd Qu.: 1466 3rd Qu.: 1234
## Max. :402701 Max. :438811 Max. :558984 Max. :427295
## Hoxa1KN_2 Hoxa1KN_3
## Min. : 0 Min. : 4
## 1st Qu.: 245 1st Qu.: 284
## Median : 587 Median : 683
## Mean : 1723 Mean : 1992
## 3rd Qu.: 1379 3rd Qu.: 1624
## Max. :498587 Max. :527743
```

## 3 Deseq 2

### 3.1 Generate result

```
colna <- colnames(cuttedData)
coldata <- as.data.frame(c('control','control','control','Hoxa1KN','Hoxa1KN','Hoxa1KN'))
colnames(coldata) <- c('condition')
```

```
rownames(coldata) <- colna
```

```
head(coldata)
```

```
##           condition
## control_1  control
## control_2  control
## control_3  control
## Hoxa1KN_1  Hoxa1KN
## Hoxa1KN_2  Hoxa1KN
## Hoxa1KN_3  Hoxa1KN
```

```
head(cuttedData)
```

```
##           control_1 control_2 control_3 Hoxa1KN_1 Hoxa1KN_2
## ENSG00000198888      17528      23007      30241      24418      29152
## ENSG00000198763      21264      26720      35550      28878      32416
## ENSG00000198804     130975     151207     195514     178130     196727
## ENSG00000198712      49769      61906      78608      66478      69758
## ENSG00000228253       9304      11160      12830      12608      13041
## ENSG00000198899      45401      51260      66851      63433      66123
##           Hoxa1KN_3
## ENSG00000198888      34416
## ENSG00000198763      38422
## ENSG00000198804     244670
## ENSG00000198712      86808
## ENSG00000228253      16063
## ENSG00000198899      79215
```

```
dds <- DESeq2::DESeqDataSetFromMatrix(countData = as.matrix(cuttedData), colData = coldata, design = ~ condition)
```

```
dds
```

```
## class: DESeqDataSet
## dim: 10859 6
## metadata(1): version
## assays(1): counts
## rownames(10859): ENSG00000198888 ENSG00000198763 ...
##      ENSG00000103222 ENSG00000048828
## rowData names(0):
## colnames(6): control_1 control_2 ... Hoxa1KN_2 Hoxa1KN_3
## colData names(1): condition
```

```
dds <- DESeq2::DESeq(dds)
```

```
## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
```

```
res <- DESeq2::results(dds)
res
```

```
## log2 fold change (MLE): condition Hoxa1KN vs control
## Wald test p-value: condition Hoxa1KN vs control
## DataFrame with 10859 rows and 6 columns
##           baseMean      log2FoldChange      lfcSE
##           <numeric>      <numeric>      <numeric>
## ENSG00000198888 25904.6473430736 0.161124364094056 0.0549351104971018
## ENSG00000198763 29989.1626645948 0.103849055295708 0.0473593152192017
## ENSG00000198804 179554.339318949 0.212049138544963 0.0363782524606112
## ENSG00000198712 67814.0020428773 0.0731078325330875 0.0432079292898051
## ENSG00000228253 12341.1879305687 0.158626546883101 0.0441781784779912
## ...
## ENSG00000173545 837.551361122154 0.257404803543496 0.0727546063210207
## ENSG00000177119 2715.69337174232 0.324855096616299 0.050828982456208
## ENSG00000205808 139.203734703101 -0.789342411447881 0.163975329910848
## ENSG00000103222 1833.02533526154 0.419437629571403 0.0680043534358781
## ENSG00000048828 3471.06222283727 -0.462607929416545 0.0439829337436671
##           stat      pvalue
##           <numeric>      <numeric>
## ENSG00000198888 2.93299426607244 0.00335710091955791
## ENSG00000198763 2.19279047458867 0.0283224785505033
## ENSG00000198804 5.82900838281225 5.57577029318664e-09
## ENSG00000198712 1.6920003743465 0.090645902249144
## ENSG00000228253 3.59060858432917 0.000329906754191281
## ...
## ENSG00000173545 3.53798634285407 0.000403190912006971
## ENSG00000177119 6.39113908873112 1.64654522353325e-10
## ENSG00000205808 -4.81378760986205 1.48096204018437e-06
## ENSG00000103222 6.16780556507893 6.92442143716582e-10
## ENSG00000048828 -10.5178961483704 7.1451187336472e-26
##           padj
##           <numeric>
## ENSG00000198888 0.00562209543168907
## ENSG00000198763 0.0412415704966221
## ENSG00000198804 1.6130066099954e-08
## ENSG00000198712 0.120590927670501
## ENSG00000228253 0.000624224055464401
## ...
## ENSG00000173545 0.000754471515280883
## ENSG00000177119 5.26399926145481e-10
## ENSG00000205808 3.53691264194494e-06
## ENSG00000103222 2.13696542192465e-09
## ENSG00000048828 4.63685320329992e-25
```

```
summary(res)
```

```
##
## out of 10859 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 4007, 37%
## LFC < 0 (down)    : 4009, 37%
## outliers [1]      : 2, 0.018%
## low counts [2]    : 0, 0%
```

```
## (mean count < 26)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

### 3.2 Cut result by FDR = 0.05 and abs(lfc) > 1

```
res05 <- DESeq2::results(dds, alpha=0.05, lfcThreshold=1, altHypothesis = 'greaterAbs')
res05inc <- DESeq2::results(dds, alpha=0.05, lfcThreshold=1, altHypothesis = 'greater')
res05decs <- DESeq2::results(dds, alpha=0.05, lfcThreshold=1, altHypothesis = 'less')
```

```
summary(res05)
```

```
##
## out of 10859 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 1.00 (up) : 391, 3.6%
## LFC < -1.00 (down) : 516, 4.8%
## outliers [1] : 2, 0.018%
## low counts [2] : 0, 0%
## (mean count < 26)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
summary(res05inc)
```

```
##
## out of 10859 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 1.00 (up) : 390, 3.6%
## LFC < -1.00 (down) : 0, 0%
## outliers [1] : 2, 0.018%
## low counts [2] : 0, 0%
## (mean count < 26)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
summary(res05decs)
```

```
##
## out of 10859 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 1.00 (up) : 0, 0%
## LFC < -1.00 (down) : 518, 4.8%
## outliers [1] : 2, 0.018%
## low counts [2] : 0, 0%
## (mean count < 26)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

### 3.3 Get Gene List, sort by lfc

```
UpGeneFilter <- which( !is.na(res$padj) &
                      !is.na(res$log2FoldChange) &
```

```

        res$padj < 0.05 &
        res$log2FoldChange > 1)

UpGeneslfc <- res$log2FoldChange[UpGeneFilter]
UpGeneList <- rownames(res)[UpGeneFilter][order(-UpGeneslfc)]

DownGeneFilter <- which( !is.na(res$padj) &
                        !is.na(res$log2FoldChange) &
                        res$padj < 0.05 &
                        res$log2FoldChange < -1)
DownGeneslfc <- res$log2FoldChange[DownGeneFilter]
DownGeneList <- rownames(res)[DownGeneFilter][order(DownGeneslfc)]

AllGeneList <- rownames(res)

length(AllGeneList)

## [1] 10859
length(UpGeneList)

## [1] 665
length(DownGeneList)

## [1] 882

```

### 3.4 Refine hierarchical clustering

Use all differentially expressed gene

```

vsd <- vst(dds, blind=FALSE)
ntd <- normTransform(dds)
resLFC <- DESeq2::lfcShrink(dds, coef="condition_Hoxa1KN_vs_control", type="apeglm")

## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
##     sequence count data: removing the noise and preserving large differences.
##     bioRxiv. https://doi.org/10.1101/303255

select2 <- order(abs(resLFC$log2FoldChange[which(resLFC$padj<0.05&abs(resLFC$log2FoldChange)>1)]),
                 decreasing=TRUE)

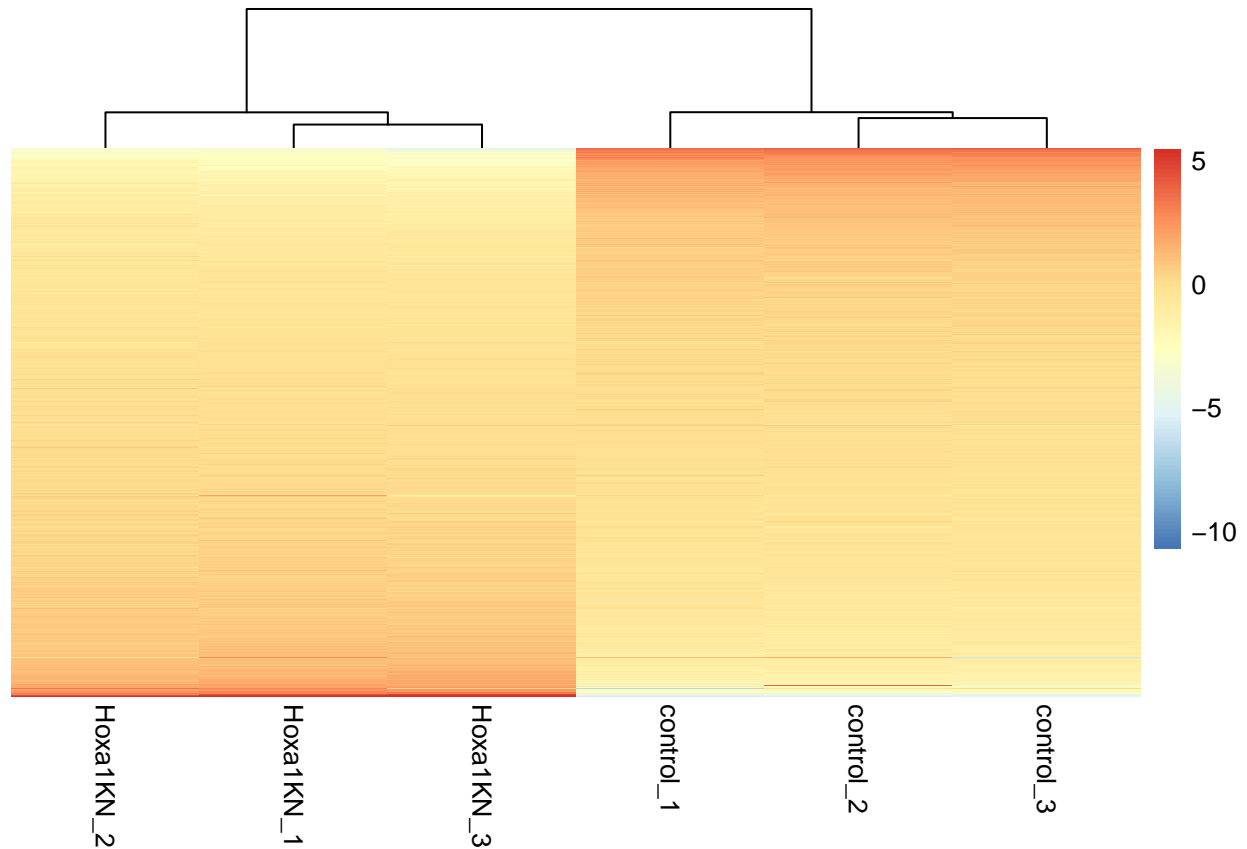
select_diff <- assay(ntd)[select2,]-rowMeans(assay(ntd)[select2,])

select3 <- order(rowMeans(select_diff[,1:3]), decreasing = TRUE)
#df <- as.data.frame(colData(dds)[,c("condition", "type")])
index <- select2[select3]
length(index)

## [1] 1494

mySd <- sd( assay(ntd)[index,]-rowMeans(assay(ntd)[index,]) )
pheatmap(( assay(ntd)[index,]-rowMeans(assay(ntd)[index,]) )/mySd, cluster_rows=FALSE, show_rownames=FALSE,
         cluster_cols=TRUE)

```



## 4. Enrichment: TopGo Solution

I failed to use topGo package

### 4.1 Load Ensembl to Go\_id table

```
# if(!file.exists('EG2GO.RDS')){
#   grch37 = useEnsembl(biomart="ensembl", dataset = "hsapiens_gene_ensembl")
#   EG2GO <- getBM(mart=grch37, attributes=c('ensembl_gene_id','go_id'))
#   saveRDS(EG2GO, file='EG2GO.RDS')
# }else{
#   EG2GO <- readRDS('EG2GO.RDS')
# }
# ...
#
# ## 4.2 Convert Ensembl Id into Go Id (or name)
# ```{r}
# EG2GO <- EG2GO[EG2GO$go_id != '',]
#
# upGoId <- EG2GO %>%
#   filter(ensembl_gene_id %in% UpGeneList) %>%
#   dplyr::select(go_id)
```

```

# upGoId <- unique(upGoId)
#
# downGoId <- EG2GO %>%
#   filter(ensembl_gene_id %in% DownGeneList) %>%
#   dplyr::select(go_id)
# downGoId <- unique(downGoId)
#
# allGoId <- EG2GO %>%
#   filter(ensembl_gene_id %in% AllGeneList) %>%
#   dplyr::select(go_id)
# allGoId <- unique(allGoId)
#
# ```
#
#
#
# ## 4.3 Using topGo package to do enrichment tests
#
# ```{r}
# xx <- annFUN.org("BP", mapping = "org.Hs.eg.db", ID = "symbol")
# head(xx)
#
# geneList <- factor(as.integer( allGoId[,1] %in% upGoId[,1]))
# names(geneList) <- allGoId
#
# str(geneList)
#
# upGoData <- new("topGOdata",
#               description='GO data for increased genes',
#               ontology = "BP", allGenes = geneList, nodeSize = 5,
#               annot = annFUN.org, mapping = "org.Hs.eg.db", ID = "symbol")
#
#
# resultFisher <- runTest(upGoData, algorithm = "classic", statistic = "fisher")
# resultFisher
#
# resultKS <- runTest(sampleGOdata, algorithm = "classic", statistic = "ks")
# resultKS.elim <- runTest(sampleGOdata, algorithm = "elim", statistic = "ks")
#
# showSigOfNodes(upGoData, score(resultKS.elim), firstSigNodes = 5, useInfo = 'all')

```

## 5. Pathway: Gage

### 5.1 Map Ids

```

res$symbol = mapIds(org.Hs.eg.db,
                    keys=row.names(res),
                    column="SYMBOL",
                    keytype="ENSEMBL",
                    multiVals="first")

```



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

```
res$entrez = mapIds(org.Hs.eg.db,  
                    keys=row.names(res),  
                    column="ENTREZID",  
                    keytype="ENSEMBL",  
                    multiVals="first")
```

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

```
res$goids = mapIds(org.Hs.eg.db,  
                   keys=row.names(res),  
                   column="GO",  
                   keytype="ENSEMBL",  
                   multiVals="first")
```

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

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

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

## 5.2 Gage analysis using kegg

```
library(gageData)  
resLFC = res$log2FoldChange  
names(resLFC) = res$entrez  
data(kegg.sets.hs)  
  
kres = gage(resLFC, gsets=kegg.sets.hs, same.dir=TRUE)  
  
k_pathways_Up = data.frame(id=rownames(kres$greater), kres$greater) %>%  
  tbl_df() %>%  
  filter(row_number() <= 5) %>%  
  .$id %>%  
  as.character()  
  
k_pathways_Down = data.frame(id=rownames(kres$less), kres$less) %>%  
  tbl_df() %>%  
  filter(row_number() <= 5) %>%  
  .$id %>%  
  as.character()  
  
k_Up_ids = substr(k_pathways_Up, start=1, stop=8)  
k_Down_ids = substr(k_pathways_Down, start=1, stop=8)
```

## Up regulated pathways

```
pathview(gene.data=resLFC, pathway.id=k_Up_ids, species="hsa", kegg.dir = "PathviewResult/Up/", new.sig

## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory C:/GitRepo/STATBioinformatics
## Info: Writing image file hsa04060.pathview.png
## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory C:/GitRepo/STATBioinformatics
## Info: Writing image file hsa04142.pathview.png
## Info: some node width is different from others, and hence adjusted!
## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory C:/GitRepo/STATBioinformatics
## Info: Writing image file hsa05323.pathview.png
## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory C:/GitRepo/STATBioinformatics
## Info: Writing image file hsa04974.pathview.png
## Info: some node width is different from others, and hence adjusted!
## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory C:/GitRepo/STATBioinformatics
## Info: Writing image file hsa05146.pathview.png
for(f in k_Up_ids){
  f <- paste0(f, '.pathview.png')
  knitr::include_graphics(f)
}
```

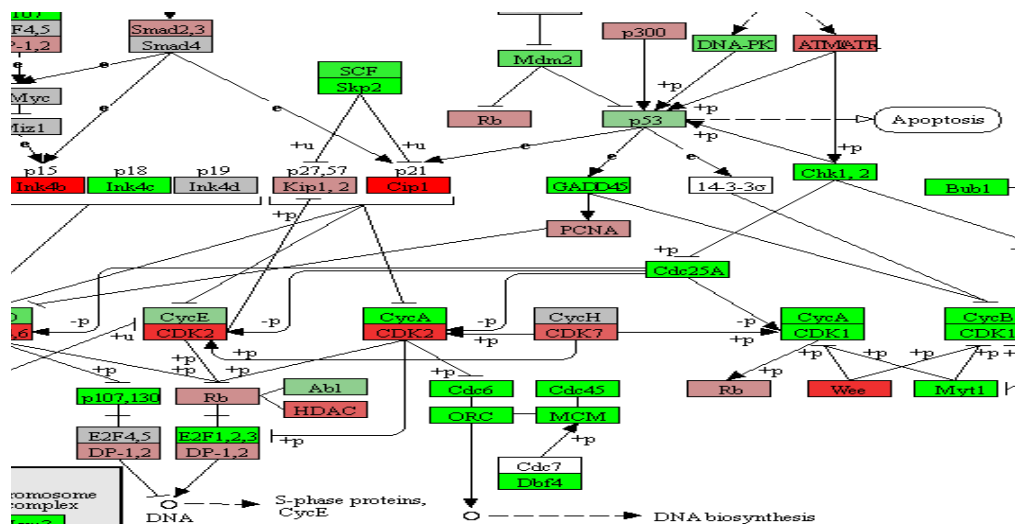
## Down regulated pathways

```
pathview(gene.data=resLFC, pathway.id=k_Down_ids, species="hsa", kegg.dir = "PathviewResult/Down/", new

## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory C:/GitRepo/STATBioinformatics
## Info: Writing image file hsa04110.pathview.png
## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory C:/GitRepo/STATBioinformatics
## Info: Writing image file hsa03030.pathview.png
## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory C:/GitRepo/STATBioinformatics
## Info: Writing image file hsa04114.pathview.png
```

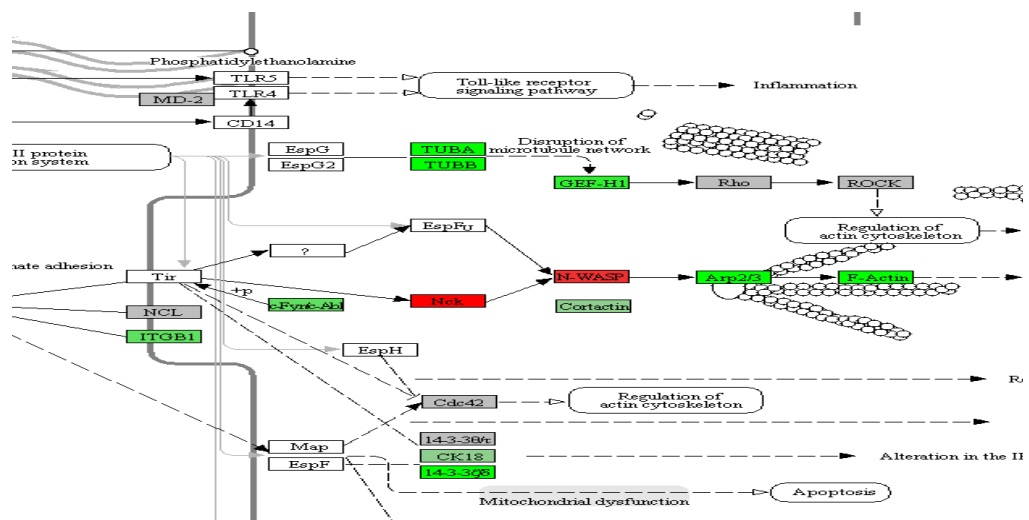
```
## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory C:/GitRepo/STATBioinformatics
## Info: Writing image file hsa05130.pathview.png
## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory C:/GitRepo/STATBioinformatics
## Info: Writing image file hsa03440.pathview.png
```

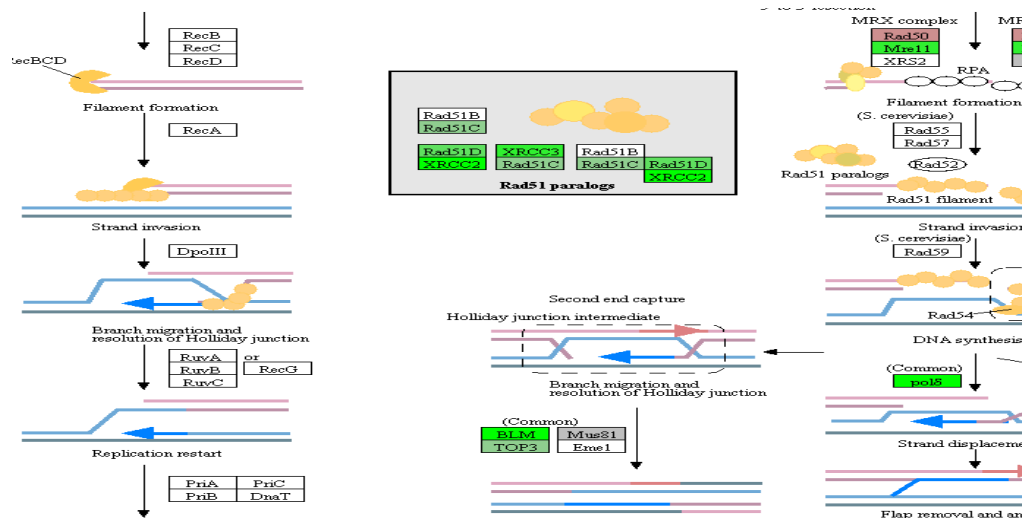
```
for(f in k_Down_ids){
  f <- paste0(f, '.pathview.png')
  img <- readPNG(f)
  plot.new()
  rasterImage(img, -0.5, -0.3, 1.5, 1.3)
}
```











### 5.3 Gage analysis using GO

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

gobpsets = go.sets.hs[go.subs.hs$BP]
goccsets = go.sets.hs[go.subs.hs$CC]
gomfsets = go.sets.hs[go.subs.hs$MF]

GoLFC <- resLFC
names(GoLFC) <- res$goids

gobpres = gage(GoLFC, gsets=gobpsets, same.dir=TRUE)
gocpres = gage(GoLFC, gsets=goccsets, same.dir=TRUE)
gomfres = gage(GoLFC, gsets=gomfsets, same.dir=TRUE)
```

bp result

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

```
## $greater
##
## GO:0000002 mitochondrial genome maintenance      p.geomean stat.mean p.val
##                                                    NA      NaN      NA
```

```

## G0:0000003 reproduction NA NaN NA
## G0:0000012 single strand break repair NA NaN NA
## G0:0000018 regulation of DNA recombination NA NaN NA
## G0:0000019 regulation of mitotic recombination NA NaN NA
## G0:0000022 mitotic spindle elongation NA NaN NA
##
## q.val set.size exp1
## G0:0000002 mitochondrial genome maintenance NA 0 NA
## G0:0000003 reproduction NA 0 NA
## G0:0000012 single strand break repair NA 0 NA
## G0:0000018 regulation of DNA recombination NA 0 NA
## G0:0000019 regulation of mitotic recombination NA 0 NA
## G0:0000022 mitotic spindle elongation NA 0 NA
##
## $less
##
## p.geomean stat.mean p.val
## G0:0000002 mitochondrial genome maintenance NA NaN NA
## G0:0000003 reproduction NA NaN NA
## G0:0000012 single strand break repair NA NaN NA
## G0:0000018 regulation of DNA recombination NA NaN NA
## G0:0000019 regulation of mitotic recombination NA NaN NA
## G0:0000022 mitotic spindle elongation NA NaN NA
##
## q.val set.size exp1
## G0:0000002 mitochondrial genome maintenance NA 0 NA
## G0:0000003 reproduction NA 0 NA
## G0:0000012 single strand break repair NA 0 NA
## G0:0000018 regulation of DNA recombination NA 0 NA
## G0:0000019 regulation of mitotic recombination NA 0 NA
## G0:0000022 mitotic spindle elongation NA 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

```

cc result

```
lapply(goccrs, head)
```

```

## $greater
##
## p.geomean
## G0:0000015 phosphopyruvate hydratase complex NA
## G0:0000109 nucleotide-excision repair complex NA
## G0:0000110 nucleotide-excision repair factor 1 complex NA
## G0:0000118 histone deacetylase complex NA
## G0:0000120 RNA polymerase I transcription factor complex NA
## G0:0000123 histone acetyltransferase complex NA
##
## stat.mean p.val
## G0:0000015 phosphopyruvate hydratase complex NaN NA
## G0:0000109 nucleotide-excision repair complex NaN NA

```



```

## G0:0000110 nucleotide-excision repair factor 1 complex      NaN    NA
## G0:0000118 histone deacetylase complex                      NaN    NA
## G0:0000120 RNA polymerase I transcription factor complex    NaN    NA
## G0:0000123 histone acetyltransferase complex                NaN    NA
##                                                              q.val set.size
## G0:0000015 phosphopyruvate hydratase complex              NA      0
## G0:0000109 nucleotide-excision repair complex              NA      0
## G0:0000110 nucleotide-excision repair factor 1 complex      NA      0
## G0:0000118 histone deacetylase complex                      NA      0
## G0:0000120 RNA polymerase I transcription factor complex    NA      0
## G0:0000123 histone acetyltransferase complex                NA      0
##                                                              exp1
## G0:0000015 phosphopyruvate hydratase complex              NA
## G0:0000109 nucleotide-excision repair complex              NA
## G0:0000110 nucleotide-excision repair factor 1 complex      NA
## G0:0000118 histone deacetylase complex                      NA
## G0:0000120 RNA polymerase I transcription factor complex    NA
## G0:0000123 histone acetyltransferase complex                NA
##
## $less
##                                                              p.geomean
## G0:0000015 phosphopyruvate hydratase complex              NA
## G0:0000109 nucleotide-excision repair complex              NA
## G0:0000110 nucleotide-excision repair factor 1 complex      NA
## G0:0000118 histone deacetylase complex                      NA
## G0:0000120 RNA polymerase I transcription factor complex    NA
## G0:0000123 histone acetyltransferase complex                NA
##                                                              stat.mean p.val
## G0:0000015 phosphopyruvate hydratase complex              NaN    NA
## G0:0000109 nucleotide-excision repair complex              NaN    NA
## G0:0000110 nucleotide-excision repair factor 1 complex      NaN    NA
## G0:0000118 histone deacetylase complex                      NaN    NA
## G0:0000120 RNA polymerase I transcription factor complex    NaN    NA
## G0:0000123 histone acetyltransferase complex                NaN    NA
##                                                              q.val set.size
## G0:0000015 phosphopyruvate hydratase complex              NA      0
## G0:0000109 nucleotide-excision repair complex              NA      0
## G0:0000110 nucleotide-excision repair factor 1 complex      NA      0
## G0:0000118 histone deacetylase complex                      NA      0
## G0:0000120 RNA polymerase I transcription factor complex    NA      0
## G0:0000123 histone acetyltransferase complex                NA      0
##                                                              exp1
## G0:0000015 phosphopyruvate hydratase complex              NA
## G0:0000109 nucleotide-excision repair complex              NA
## G0:0000110 nucleotide-excision repair factor 1 complex      NA
## G0:0000118 histone deacetylase complex                      NA
## G0:0000120 RNA polymerase I transcription factor complex    NA
## G0:0000123 histone acetyltransferase complex                NA
##
## $stats
##                                                              stat.mean exp1
## G0:0000015 phosphopyruvate hydratase complex              NaN    NA
## G0:0000109 nucleotide-excision repair complex              NaN    NA
## G0:0000110 nucleotide-excision repair factor 1 complex      NaN    NA

```

## GO:0000118 histone deacetylase complex	NaN	NA
## GO:0000120 RNA polymerase I transcription factor complex	NaN	NA
## GO:0000123 histone acetyltransferase complex	NaN	NA

mf result

```
lapply(gomfres, head)
```

```
## $greater
##
##                                p.geomean
## GO:0000009 alpha-1,6-mannosyltransferase activity      NA
## GO:0000010 trans-hexaprenyltranstransferase activity  NA
## GO:0000014 single-stranded DNA specific endodeoxyribonuclease activity  NA
## GO:0000016 lactase activity                             NA
## GO:0000026 alpha-1,2-mannosyltransferase activity      NA
## GO:0000030 mannosyltransferase activity                NA
##                                stat.mean
## GO:0000009 alpha-1,6-mannosyltransferase activity      NaN
## GO:0000010 trans-hexaprenyltranstransferase activity  NaN
## GO:0000014 single-stranded DNA specific endodeoxyribonuclease activity  NaN
## GO:0000016 lactase activity                             NaN
## GO:0000026 alpha-1,2-mannosyltransferase activity      NaN
## GO:0000030 mannosyltransferase activity                NaN
##                                p.val
## GO:0000009 alpha-1,6-mannosyltransferase activity      NA
## GO:0000010 trans-hexaprenyltranstransferase activity  NA
## GO:0000014 single-stranded DNA specific endodeoxyribonuclease activity  NA
## GO:0000016 lactase activity                             NA
## GO:0000026 alpha-1,2-mannosyltransferase activity      NA
## GO:0000030 mannosyltransferase activity                NA
##                                q.val
## GO:0000009 alpha-1,6-mannosyltransferase activity      NA
## GO:0000010 trans-hexaprenyltranstransferase activity  NA
## GO:0000014 single-stranded DNA specific endodeoxyribonuclease activity  NA
## GO:0000016 lactase activity                             NA
## GO:0000026 alpha-1,2-mannosyltransferase activity      NA
## GO:0000030 mannosyltransferase activity                NA
##                                set.size
## GO:0000009 alpha-1,6-mannosyltransferase activity      0
## GO:0000010 trans-hexaprenyltranstransferase activity  0
## GO:0000014 single-stranded DNA specific endodeoxyribonuclease activity  0
## GO:0000016 lactase activity                             0
## GO:0000026 alpha-1,2-mannosyltransferase activity      0
## GO:0000030 mannosyltransferase activity                0
##                                exp1
## GO:0000009 alpha-1,6-mannosyltransferase activity      NA
## GO:0000010 trans-hexaprenyltranstransferase activity  NA
## GO:0000014 single-stranded DNA specific endodeoxyribonuclease activity  NA
## GO:0000016 lactase activity                             NA
## GO:0000026 alpha-1,2-mannosyltransferase activity      NA
## GO:0000030 mannosyltransferase activity                NA
##
## $less
```

```

## p.geomean
## G0:0000009 alpha-1,6-mannosyltransferase activity NA
## G0:0000010 trans-hexaprenyltranstransferase activity NA
## G0:0000014 single-stranded DNA specific endodeoxyribonuclease activity NA
## G0:0000016 lactase activity NA
## G0:0000026 alpha-1,2-mannosyltransferase activity NA
## G0:0000030 mannosyltransferase activity NA
## stat.mean
## G0:0000009 alpha-1,6-mannosyltransferase activity NaN
## G0:0000010 trans-hexaprenyltranstransferase activity NaN
## G0:0000014 single-stranded DNA specific endodeoxyribonuclease activity NaN
## G0:0000016 lactase activity NaN
## G0:0000026 alpha-1,2-mannosyltransferase activity NaN
## G0:0000030 mannosyltransferase activity NaN
## p.val
## G0:0000009 alpha-1,6-mannosyltransferase activity NA
## G0:0000010 trans-hexaprenyltranstransferase activity NA
## G0:0000014 single-stranded DNA specific endodeoxyribonuclease activity NA
## G0:0000016 lactase activity NA
## G0:0000026 alpha-1,2-mannosyltransferase activity NA
## G0:0000030 mannosyltransferase activity NA
## q.val
## G0:0000009 alpha-1,6-mannosyltransferase activity NA
## G0:0000010 trans-hexaprenyltranstransferase activity NA
## G0:0000014 single-stranded DNA specific endodeoxyribonuclease activity NA
## G0:0000016 lactase activity NA
## G0:0000026 alpha-1,2-mannosyltransferase activity NA
## G0:0000030 mannosyltransferase activity NA
## set.size
## G0:0000009 alpha-1,6-mannosyltransferase activity 0
## G0:0000010 trans-hexaprenyltranstransferase activity 0
## G0:0000014 single-stranded DNA specific endodeoxyribonuclease activity 0
## G0:0000016 lactase activity 0
## G0:0000026 alpha-1,2-mannosyltransferase activity 0
## G0:0000030 mannosyltransferase activity 0
## exp1
## G0:0000009 alpha-1,6-mannosyltransferase activity NA
## G0:0000010 trans-hexaprenyltranstransferase activity NA
## G0:0000014 single-stranded DNA specific endodeoxyribonuclease activity NA
## G0:0000016 lactase activity NA
## G0:0000026 alpha-1,2-mannosyltransferase activity NA
## G0:0000030 mannosyltransferase activity NA
##
## $stats
## stat.mean
## G0:0000009 alpha-1,6-mannosyltransferase activity NaN
## G0:0000010 trans-hexaprenyltranstransferase activity NaN
## G0:0000014 single-stranded DNA specific endodeoxyribonuclease activity NaN
## G0:0000016 lactase activity NaN
## G0:0000026 alpha-1,2-mannosyltransferase activity NaN
## G0:0000030 mannosyltransferase activity NaN
## exp1
## G0:0000009 alpha-1,6-mannosyltransferase activity NA
## G0:0000010 trans-hexaprenyltranstransferase activity NA

```

## G0:0000014	single-stranded DNA specific endodeoxyribonuclease activity	NA
## G0:0000016	lactase activity	NA
## G0:0000026	alpha-1,2-mannosyltransferase activity	NA
## G0:0000030	mannosyltransferase activity	NA