

# class16

Kai Movellan

11/18/2021

##Section 1. Differential Expression Analysis

```
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, 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 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, colAvgPerRowSet, 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, rowAvgPerColSet,
##   rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##   rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##   rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##   rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##   rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##   rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##   rowWeightedSds, rowWeightedVars

## Loading required package: Biobase

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

##
## Attaching package: 'Biobase'

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

## The following objects are masked from 'package:matrixStats':
##
##   anyMissing, rowMedians

metaFile <- "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
```

```
# 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

```
# Note we need to remove the odd first $length col
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).

Tip: What will rowSums() of countData return and how could you use it in this context

```
# Filter count data where you have 0 read count across all samples.
countnozero <- countData[rowSums(countData) != 0, ]
head(countnozero)
```

```
##                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
```

```
nrow(countnozero)
```

```
## [1] 15975
```

Running DESeq2

```
dds = DESeqDataSetFromMatrix(countData=countData,
                              colData=colData,
                              design=~condition)
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: 19808 6
## metadata(1): version
## assays(4): counts mu H cooks
## rownames(19808): ENSG00000186092 ENSG00000279928 ... ENSG00000277475
##      ENSG00000268674
## 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"))
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>
```

```
## ENSG00000186092    0.0000          NA          NA          NA          NA
## ENSG00000279928    0.0000          NA          NA          NA          NA
## ENSG00000279457    29.9136    0.179257  0.324822  0.551863  0.58104205
## ENSG00000278566    0.0000          NA          NA          NA          NA
## ENSG00000273547    0.0000          NA          NA          NA          NA
## ENSG00000187634   183.2296    0.426457  0.140266  3.040350  0.00236304
##                padj
##                <numeric>
## ENSG00000186092          NA
## ENSG00000279928          NA
## ENSG00000279457  0.68707978
## ENSG00000278566          NA
## ENSG00000273547          NA
## ENSG00000187634  0.00516278
```

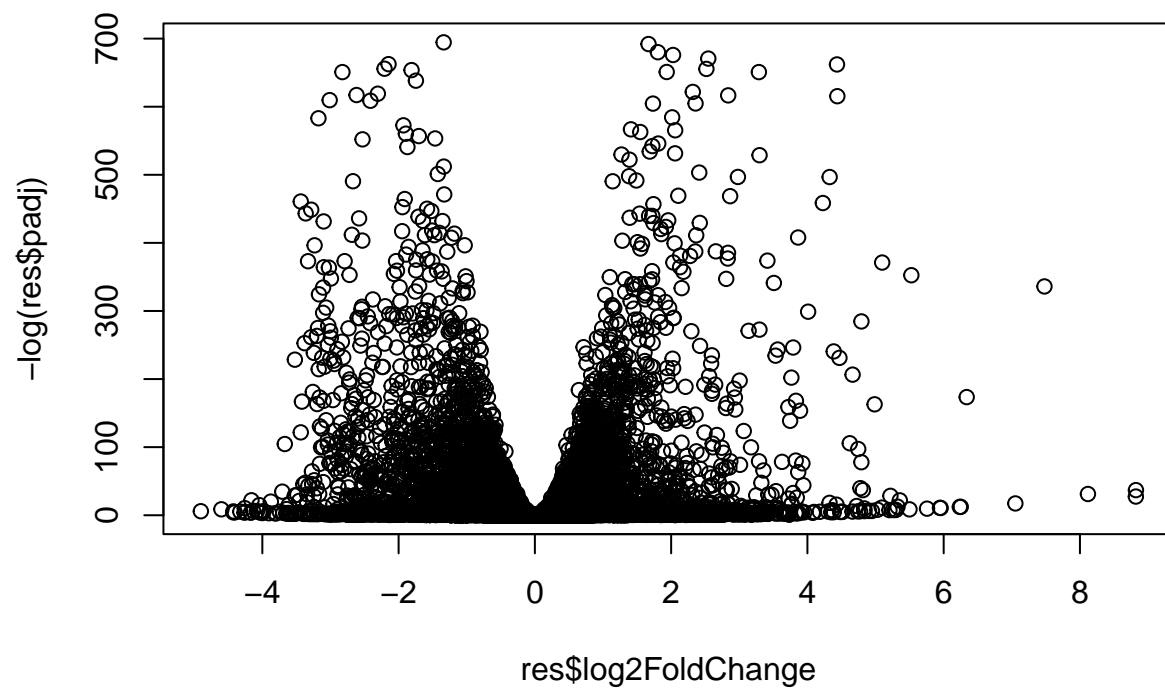
Q. Call the `summary()` function on your results to get a sense of how many genes are up or down-regulated at the default 0.1 p-value cutoff.

```
summary(res)
```

```
##
## out of 15975 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 4349, 27%
## LFC < 0 (down)     : 4393, 27%
## outliers [1]       : 0, 0%
## low counts [2]     : 1221, 7.6%
## (mean count < 0)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Volcano plot

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

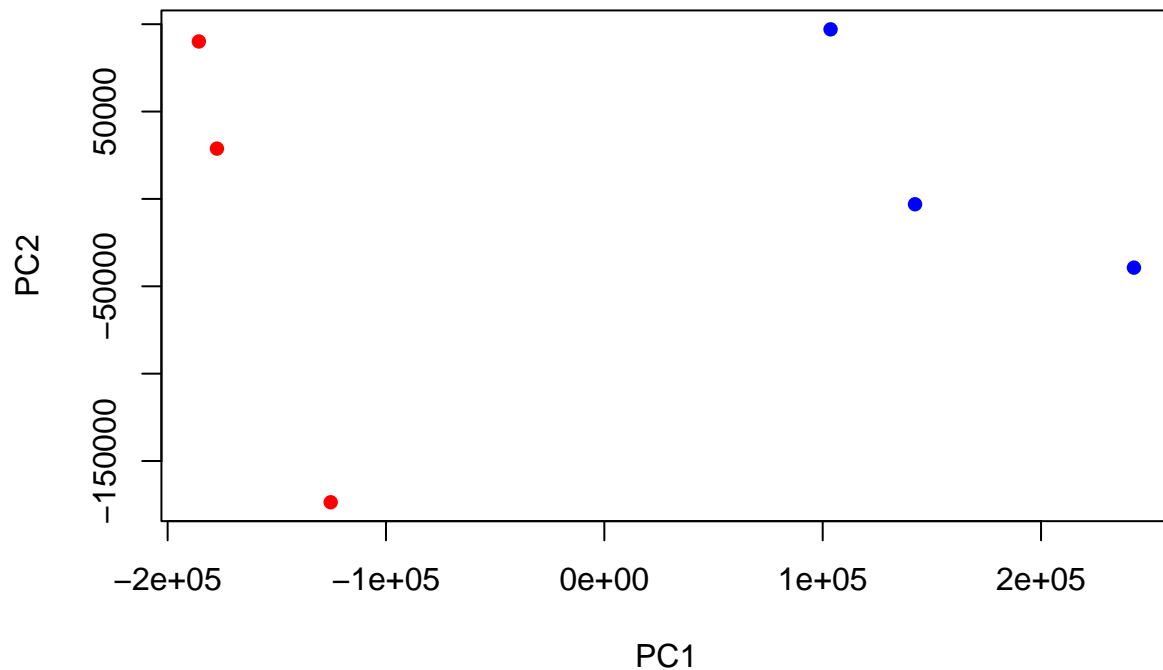


```
pca<-prcomp(t(countnozero))
```

```
mycols <- rep(c("red","blue"), each=3)  
mycols
```

```
## [1] "red" "red" "red" "blue" "blue" "blue"
```

```
plot(pca$x[,1:2], col=mycols, pch=16)
```



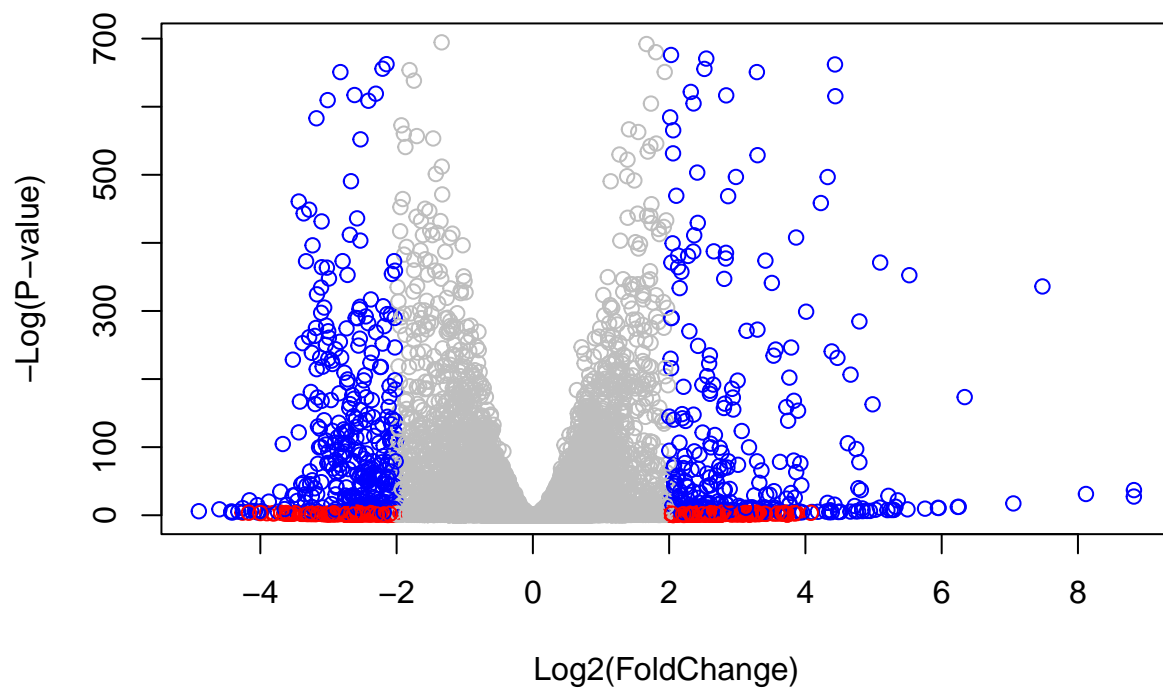
Q. Improve this plot by completing the below code, which adds color and axis labels

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

# 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$pvalue < 0.01) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

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



```
library(EnhancedVolcano)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: ggrepel
```

```
## Registered S3 methods overwritten by 'ggalt':
```

```
##   method                      from
##   grid.draw.absoluteGrob      ggplot2
##   grobHeight.absoluteGrob     ggplot2
##   grobWidth.absoluteGrob      ggplot2
##   grobX.absoluteGrob          ggplot2
##   grobY.absoluteGrob          ggplot2
```

Adding gene annotation

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>
## ENSG00000186092    0.0000          NA      NA          NA          NA
## ENSG00000279928    0.0000          NA      NA          NA          NA
## ENSG00000279457   29.9136    0.1792571 0.3248216    0.551863 5.81042e-01
## ENSG00000278566    0.0000          NA      NA          NA          NA
## ENSG00000273547    0.0000          NA      NA          NA          NA
## ENSG00000187634   183.2296    0.4264571 0.1402658    3.040350 2.36304e-03
## ENSG00000188976  1651.1881   -0.6927205 0.0548465   -12.630158 1.43990e-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 0.5215598    1.040744 2.97994e-01
```

##	padj	symbol	entrez	name
##	<numeric>	<character>	<character>	<character>
## ENSG00000186092	NA	OR4F5	79501	olfactory receptor f..
## ENSG00000279928	NA	NA	NA	NA
## ENSG00000279457	6.87080e-01	WASH9P	102723897	WAS protein family h..
## ENSG00000278566	NA	NA	NA	NA
## ENSG00000273547	NA	NA	NA	NA
## ENSG00000187634	5.16278e-03	SAMD11	148398	sterile alpha motif ..
## ENSG00000188976	1.76741e-35	NOC2L	26155	NOC2 like nucleolar ..
## ENSG00000187961	1.13536e-07	KLHL17	339451	kelch like family me..
## ENSG00000187583	9.18988e-01	PLEKHN1	84069	pleckstrin homology ..
## ENSG00000187642	4.03817e-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$pvalue),]
ord <- order( res$padj )
#View(res[ord,])
head(res[ord,])
```

```
## 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

```
write.csv(res[ord,], "deseq_results.csv")
```

##Section 2. Pathway Analysis

KEGG pathways

```
# Run in your R console (i.e. not your Rmarkdown doc!)
#BiocManager::install( c("pathview", "gage", "gageData") )

# For old versions of R only (R < 3.5.0)!
#source("http://bioconductor.org/biocLite.R")
#biocLite( c("pathview", "gage", "gageData") )
```

Now we can load the packages and setup the KEGG data-sets we need.

```
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      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)
```

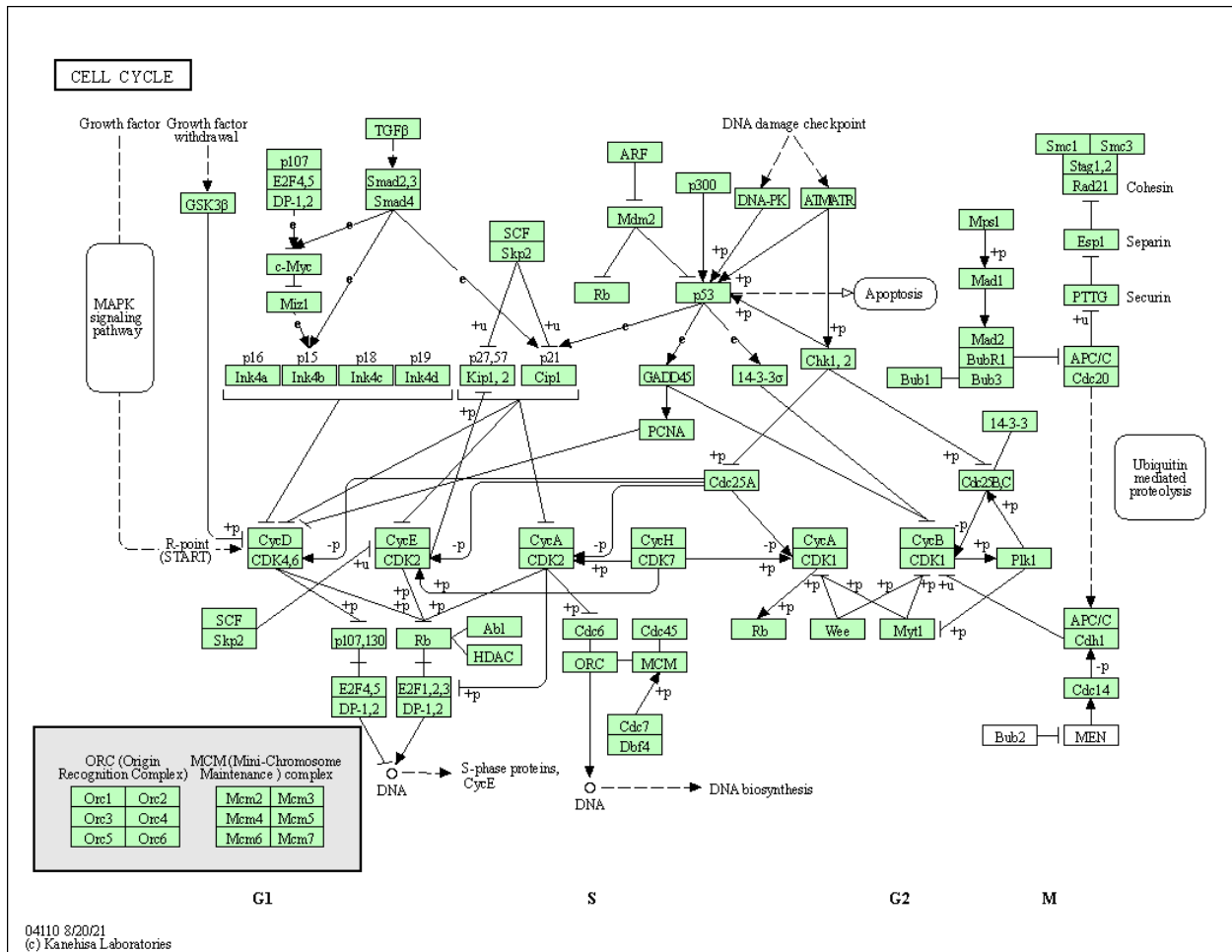
```
##                p.geomean stat.mean      p.val
## hsa04110 Cell cycle      7.077982e-06 -4.432593 7.077982e-06
## hsa03030 DNA replication  9.424076e-05 -3.951803 9.424076e-05
## hsa03013 RNA transport   1.121279e-03 -3.090949 1.121279e-03
## hsa04114 Oocyte meiosis  2.563806e-03 -2.827297 2.563806e-03
## hsa03440 Homologous recombination 3.066756e-03 -2.852899 3.066756e-03
## hsa00010 Glycolysis / Gluconeogenesis 4.360092e-03 -2.663825 4.360092e-03
##                q.val set.size      exp1
## hsa04110 Cell cycle      0.001160789      124 7.077982e-06
## hsa03030 DNA replication  0.007727742       36 9.424076e-05
## hsa03013 RNA transport   0.061296597      150 1.121279e-03
## hsa04114 Oocyte meiosis  0.100589607      112 2.563806e-03
## hsa03440 Homologous recombination 0.100589607       28 3.066756e-03
## hsa00010 Glycolysis / Gluconeogenesis 0.119175854       65 4.360092e-03
```

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

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

```
## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143_g
```

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



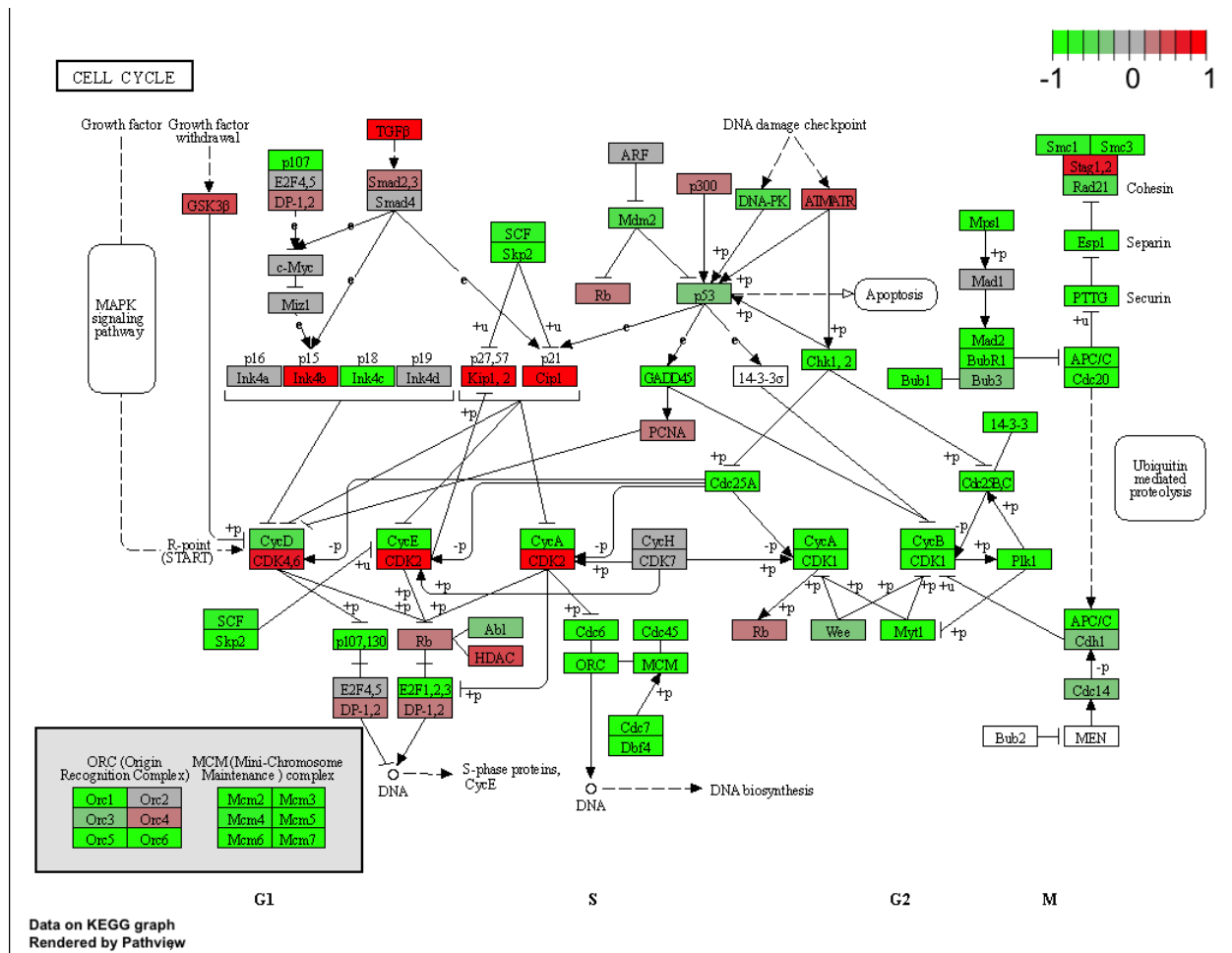
# A different PDF based output of the same data

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

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

## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143\_g

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



```
## 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] "hsa04740" "hsa04640" "hsa00140" "hsa04630" "hsa04976"
```

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

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

```
## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143_g
```

```
## Info: Writing image file hsa04740.pathview.png
```

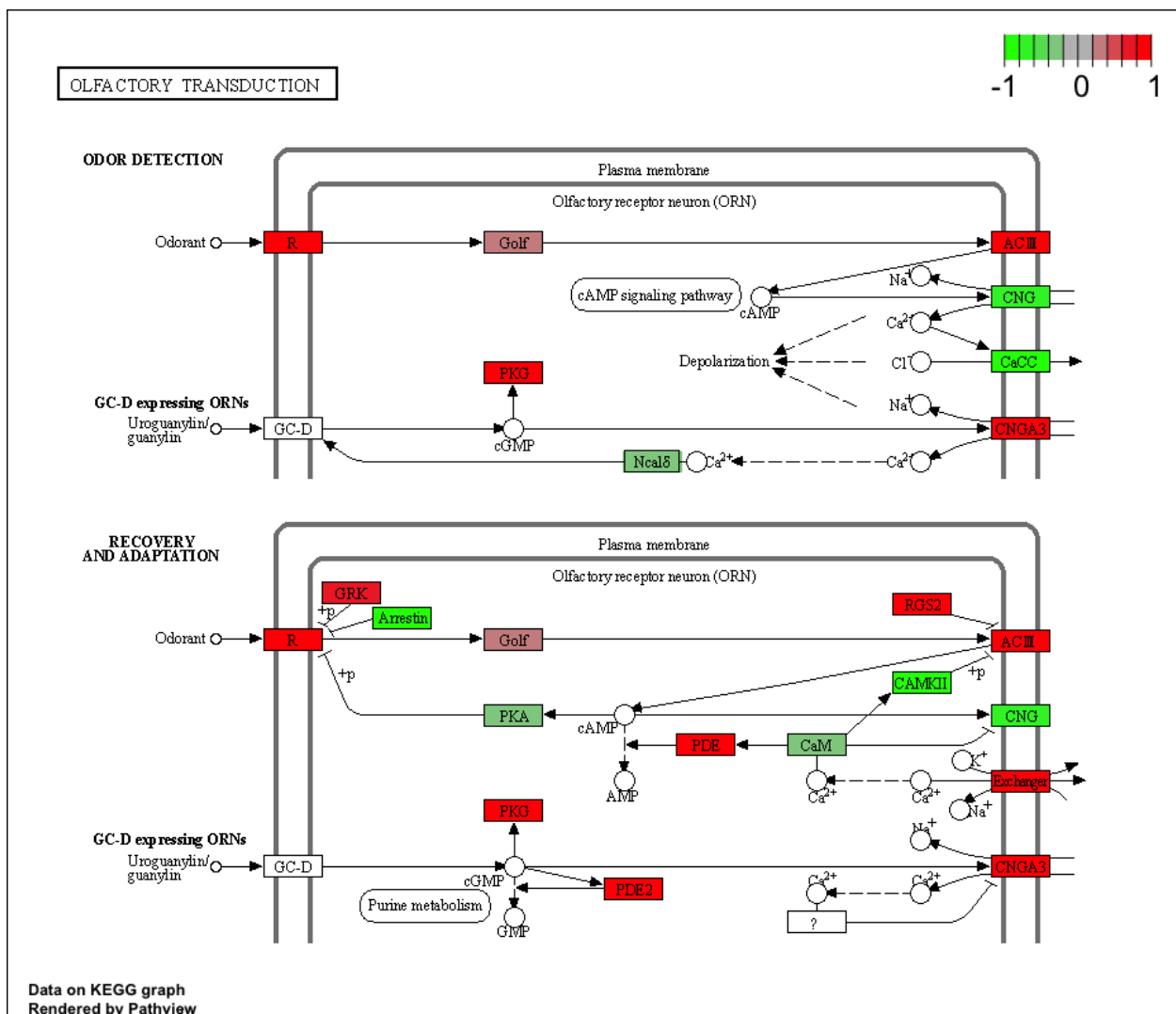
```
## Info: some node width is different from others, and hence adjusted!
```

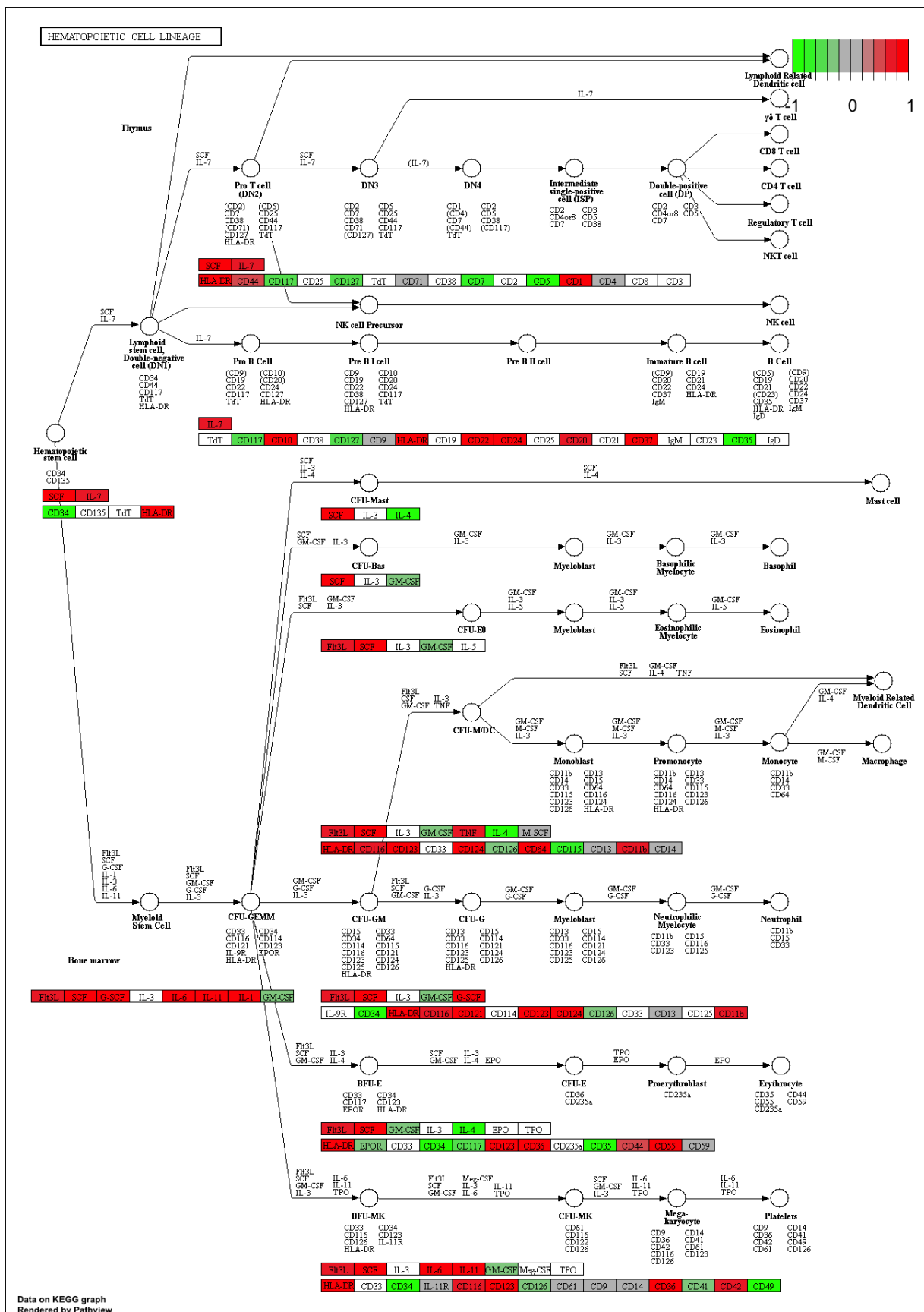
```
## 'select()' returned 1:1 mapping between keys and columns
```

```

## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143_g
## Info: Writing image file hsa04640.pathview.png
## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143_g
## Info: Writing image file hsa00140.pathview.png
## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143_g
## Info: Writing image file hsa04630.pathview.png
## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143_g
## Info: Writing image file hsa04976.pathview.png

```











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

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

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

```
## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143_g
```

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

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

```
## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143_g
```

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

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

```
## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143_g
```

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

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

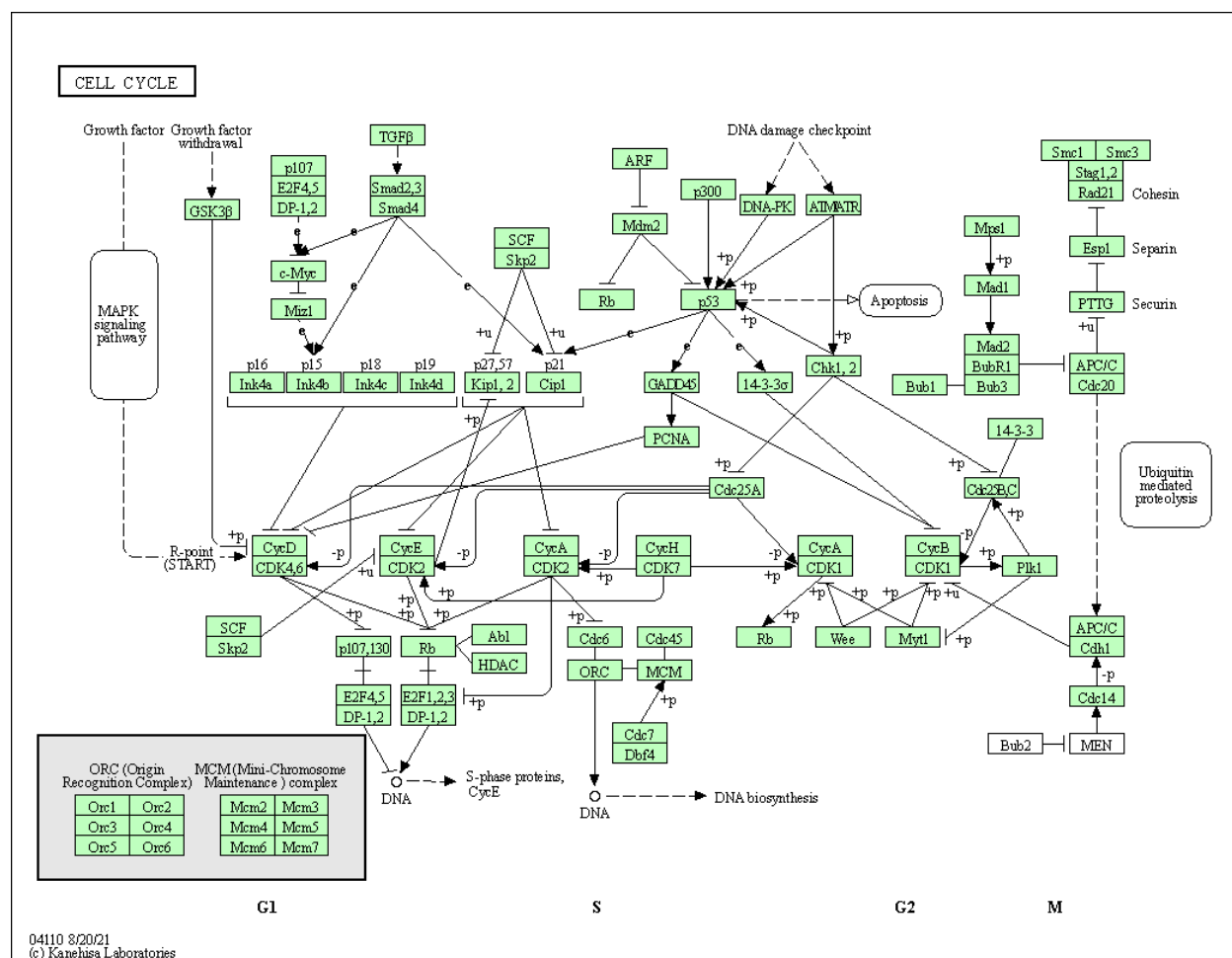
```
## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143_g
```

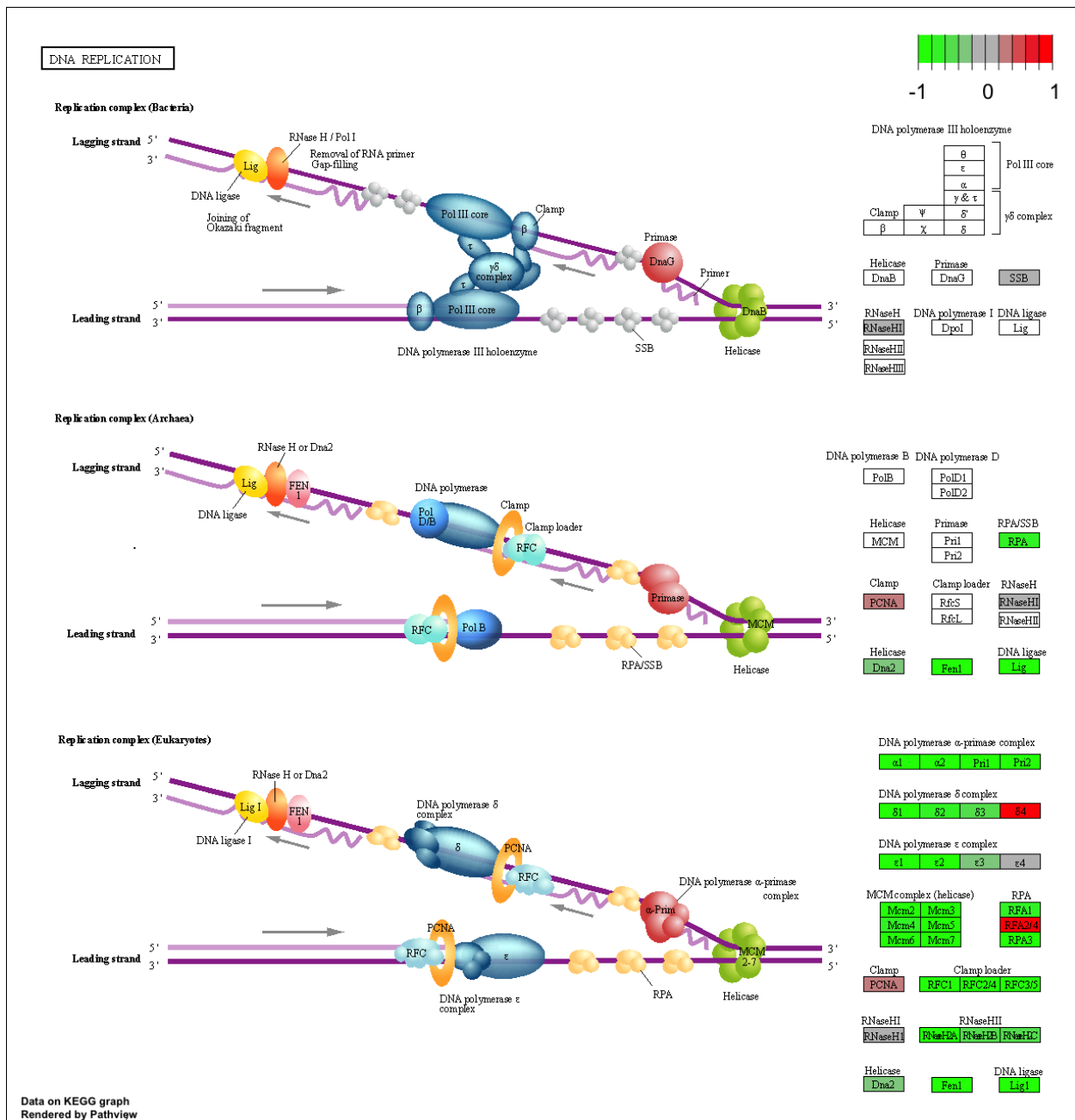
```
## Info: Writing image file hsa04114.pathview.png
```

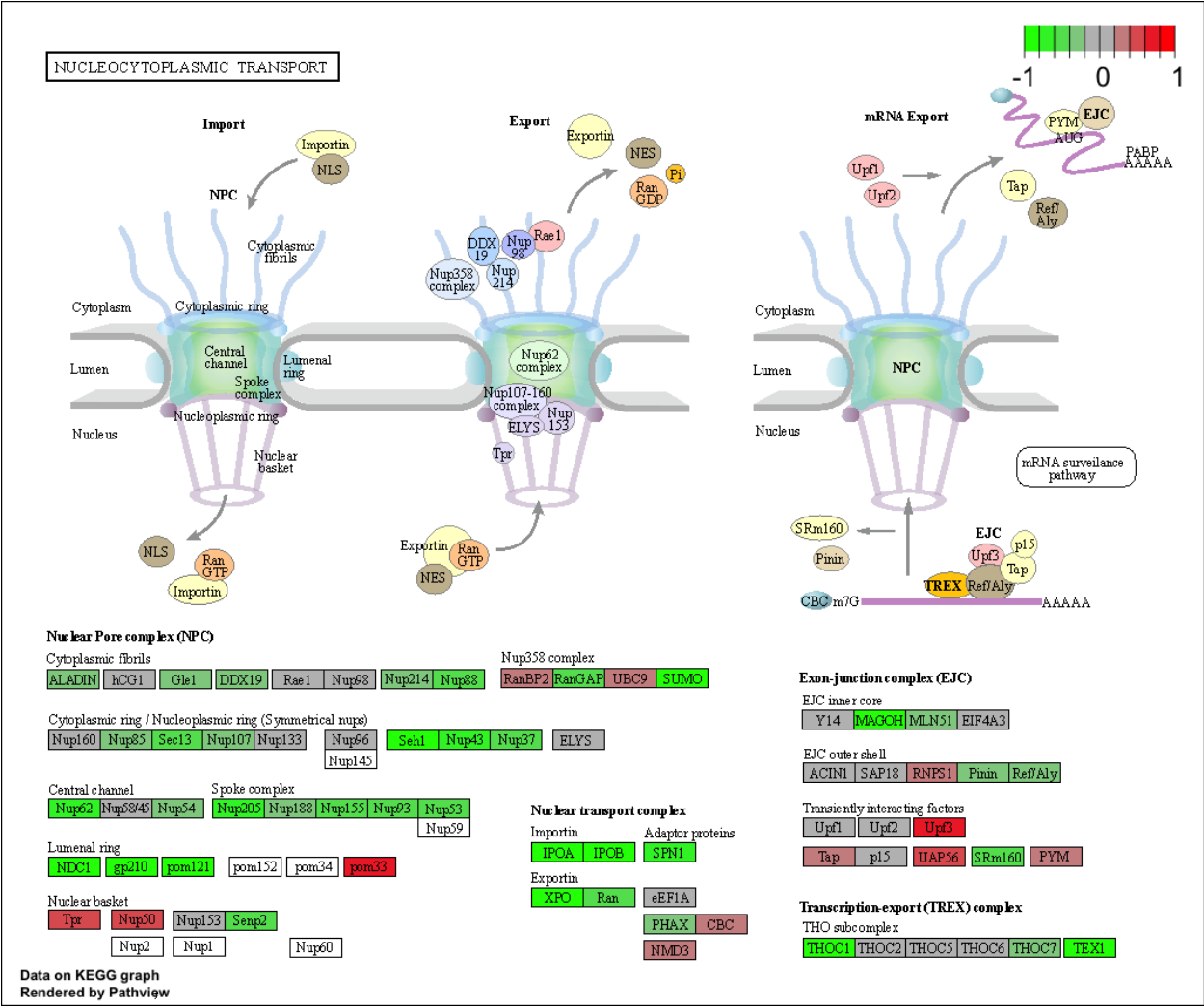
```
## 'select()' returned 1:1 mapping between keys and columns
```

```
## Info: Working in directory /Users/astro/Dropbox/My Mac (kai's MacBook Pro)/Desktop/bimm143/bimm143_g
```

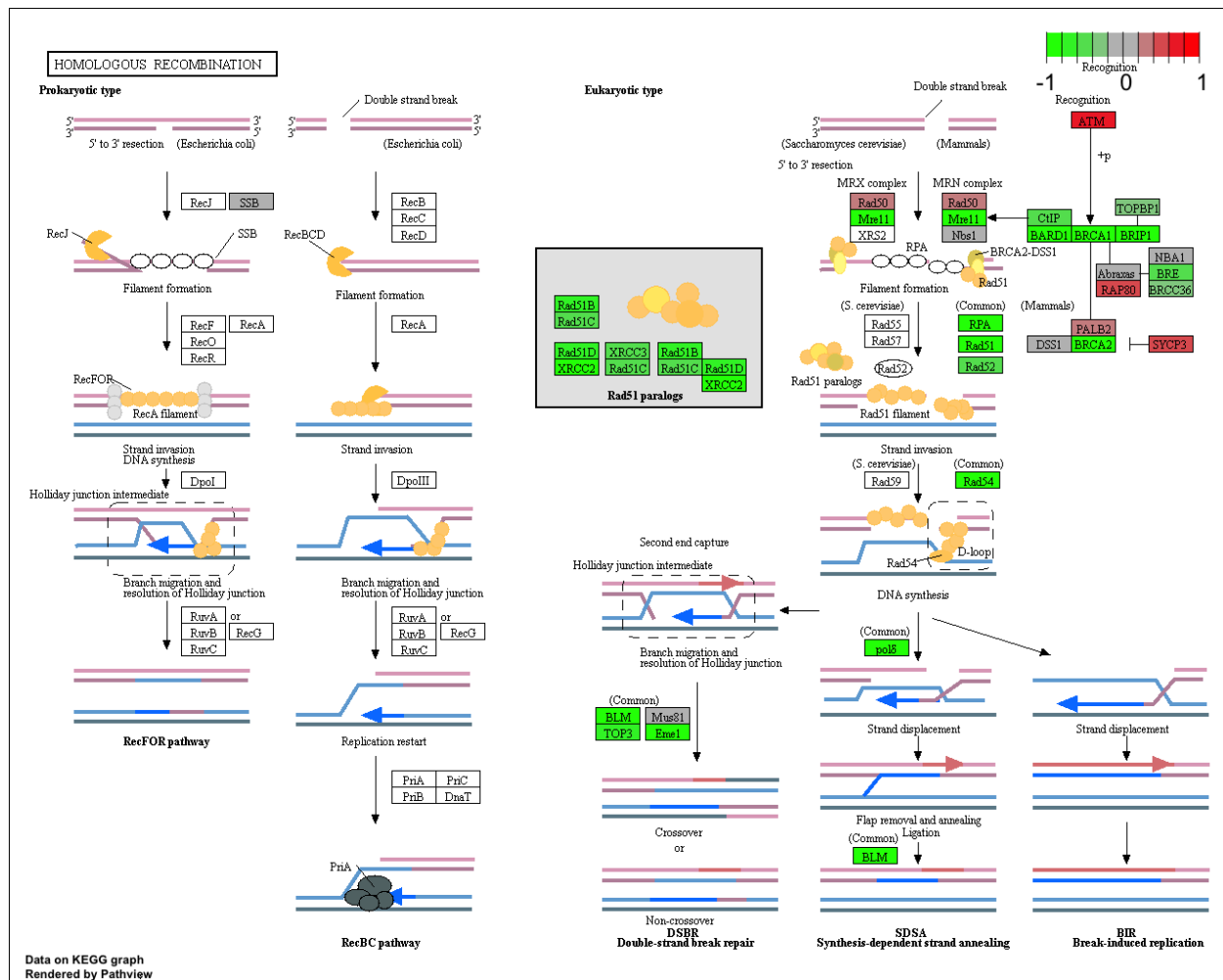
```
## Info: Writing image file hsa03440.pathview.png
```











### ##Section 3. Gene Ontology (GO)

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

```
# Focus on Biological Process subset of GO
gobpsets = go.sets.hs[go.subs.hs$BP]
```

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

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

```
## $greater
##
##      p.geomean stat.mean      p.val
## GO:0007156 homophilic cell adhesion 1.624062e-05 4.226117 1.624062e-05
## GO:0048729 tissue morphogenesis 5.407952e-05 3.888470 5.407952e-05
## GO:0002009 morphogenesis of an epithelium 5.727599e-05 3.878706 5.727599e-05
## GO:0030855 epithelial cell differentiation 2.053700e-04 3.554776 2.053700e-04
## GO:0060562 epithelial tube morphogenesis 2.927804e-04 3.458463 2.927804e-04
## GO:0048598 embryonic morphogenesis 2.959270e-04 3.446527 2.959270e-04
##
##      q.val set.size      exp1
## GO:0007156 homophilic cell adhesion 0.07103646 138 1.624062e-05
```



```
## G0:0048729 tissue morphogenesis 0.08350839 483 5.407952e-05
## G0:0002009 morphogenesis of an epithelium 0.08350839 382 5.727599e-05
## G0:0030855 epithelial cell differentiation 0.15370245 299 2.053700e-04
## G0:0060562 epithelial tube morphogenesis 0.15370245 289 2.927804e-04
## G0:0048598 embryonic morphogenesis 0.15370245 498 2.959270e-04
##
## $less
##
## p.geomean stat.mean p.val
## G0:0048285 organelle fission 6.626774e-16 -8.170439 6.626774e-16
## G0:0000280 nuclear division 1.797050e-15 -8.051200 1.797050e-15
## G0:0007067 mitosis 1.797050e-15 -8.051200 1.797050e-15
## G0:0000087 M phase of mitotic cell cycle 4.757263e-15 -7.915080 4.757263e-15
## G0:0007059 chromosome segregation 1.081862e-11 -6.974546 1.081862e-11
## G0:0051301 cell division 8.718528e-11 -6.455491 8.718528e-11
##
## q.val set.size exp1
## G0:0048285 organelle fission 2.620099e-12 386 6.626774e-16
## G0:0000280 nuclear division 2.620099e-12 362 1.797050e-15
## G0:0007067 mitosis 2.620099e-12 362 1.797050e-15
## G0:0000087 M phase of mitotic cell cycle 5.202068e-12 373 4.757263e-15
## G0:0007059 chromosome segregation 9.464127e-09 146 1.081862e-11
## G0:0051301 cell division 6.355807e-08 479 8.718528e-11
##
## $stats
##
## stat.mean exp1
## G0:0007156 homophilic cell adhesion 4.226117 4.226117
## G0:0048729 tissue morphogenesis 3.888470 3.888470
## G0:0002009 morphogenesis of an epithelium 3.878706 3.878706
## G0:0030855 epithelial cell differentiation 3.554776 3.554776
## G0:0060562 epithelial tube morphogenesis 3.458463 3.458463
## G0:0048598 embryonic morphogenesis 3.446527 3.446527
```

```
go2<- gage(foldchanges,gsets = gobpsets)
```

```
head(go2$less)
```

```
##
## p.geomean stat.mean p.val
## G0:0048285 organelle fission 6.626774e-16 -8.170439 6.626774e-16
## G0:0000280 nuclear division 1.797050e-15 -8.051200 1.797050e-15
## G0:0007067 mitosis 1.797050e-15 -8.051200 1.797050e-15
## G0:0000087 M phase of mitotic cell cycle 4.757263e-15 -7.915080 4.757263e-15
## G0:0007059 chromosome segregation 1.081862e-11 -6.974546 1.081862e-11
## G0:0051301 cell division 8.718528e-11 -6.455491 8.718528e-11
##
## q.val set.size exp1
## G0:0048285 organelle fission 2.620099e-12 386 6.626774e-16
## G0:0000280 nuclear division 2.620099e-12 362 1.797050e-15
## G0:0007067 mitosis 2.620099e-12 362 1.797050e-15
## G0:0000087 M phase of mitotic cell cycle 5.202068e-12 373 4.757263e-15
## G0:0007059 chromosome segregation 9.464127e-09 146 1.081862e-11
## G0:0051301 cell division 6.355807e-08 479 8.718528e-11
```

```
##Section 4. Reactome Analysis
```

```
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: 8146"
```

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

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?

0048285 organelle fission has the most significant p-value for the reactome result

hsa04110 Cell cycle listen on KEGG does not match the organelle fission

Difference between the two methods could be the databases they are using.

##Section 5. GO online (OPTIONAL)

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?

Endosomal/Vacuolar pathway has the most significant p-value on reactome

Endosomal pathway does not match the previous KEGG results of hsa04110 Cell cycle

Difference between the two methods could be the databases they are using.