Assignment_2

Matthew Ng

2025-10-31

```
# Load libraries
library(DESeq2)
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: generics
```

```
##
## Attaching package: 'generics'
```

```
## The following objects are masked from 'package:base':
##
##     as.difftime, as.factor, as.ordered, intersect, is.element, setdiff,
##     setequal, union
```

```
##
## 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, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, saveRDS, table, tapply, 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: Seqinfo

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: Biobase

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

##
## Attaching package: 'Biobase'
```

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

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

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:Biobase':
##
##     combine

## The following object is masked from 'package:matrixStats':
##
##     count

## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union

## The following object is masked from 'package:Seqinfo':
##
##     intersect

## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union

## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, setequal, union

## The following object is masked from 'package:generics':
##
##     explain

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(pheatmap)
library(clusterProfiler)
```

```
##
```

```
## clusterProfiler v4.18.1 Learn more at https://yulab-smu.top/contribution-knowledge-mining/
##
## Please cite:
##
## S Xu, E Hu, Y Cai, Z Xie, X Luo, L Zhan, W Tang, Q Wang, B Liu, R Wang,
## W Xie, T Wu, L Xie, G Yu. Using clusterProfiler to characterize
## multiomics data. Nature Protocols. 2024, 19(11):3292-3320
```

```
##
## Attaching package: 'clusterProfiler'
```

```
## The following object is masked from 'package:IRanges':
##
##     slice
```

```
## The following object is masked from 'package:S4Vectors':
##
##     rename
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

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

```
## Loading required package: AnnotationDbi
```

```
##
## Attaching package: 'AnnotationDbi'
```

```
## The following object is masked from 'package:clusterProfiler':
##
##     select
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
##
```

```r
# 1. Read in count data
countData <- read.delim("/Users/matthewng/Downloads/gene_counts.txt", comment.char = "#")
rownames(countData) <- countData$Geneid
countData <- countData[,7:ncol(countData)]  # keep only sample columns
```

```r
# 2. Read sample metadata
colData <- read.csv("/Users/matthewng/Downloads/assignment_2_info.csv", stringsAsFactors = TRUE)

# Match sample order
colnames(countData) <- colData$SRA.Accession

# Check
head(colData)
```

```
##    Condition Sample Time.Point SRA.Accession  X X.1 X.2 X.3 X.4 X.5
## 1      Mock       1         24   SRR22269883 NA  NA  NA  NA  NA  NA
## 2      Mock       2         24   SRR22269882 NA  NA  NA  NA  NA  NA
## 3      Mock       3         24   SRR22269881 NA  NA  NA  NA  NA  NA
## 4 SARS-CoV-2      1         24   SRR22269880 NA  NA  NA  NA  NA  NA
## 5 SARS-CoV-2      2         24   SRR22269879 NA  NA  NA  NA  NA  NA
## 6 SARS-CoV-2      3         24   SRR22269878 NA  NA  NA  NA  NA  NA
```

```r
head(countData[,1:3])
```

```
##                 SRR22269883 SRR22269882 SRR22269881
## ENSG00000223972           0           0           0
## ENSG00000227232           0           0           0
## ENSG00000278267           0           0           0
## ENSG00000243485           0           0           0
## ENSG00000274890           0           0           0
## ENSG00000237613           0           0           0
```

```r
# Prepare metadata
#colData$Condition <- factor(colData$Condition, levels = c("Mock", "SARS-CoV-2"))
#colData$Time.Point <- factor(colData$Time.Point, levels = c(24, 72))

# Check structure
str(colData)
```

```
## 'data.frame':    12 obs. of  10 variables:
##  $ Condition    : Factor w/ 2 levels "Mock ","SARS-CoV-2": 1 1 1 2 2 2 1 1 1 2 ...
##  $ Sample       : int  1 2 3 1 2 3 1 2 3 1 ...
##  $ Time.Point   : int  24 24 24 24 24 24 72 72 72 72 ...
##  $ SRA.Accession: Factor w/ 12 levels "SRR22269872",..: 12 11 10 9 8 7 6 5 4 3 ...
##  $ X            : logi  NA NA NA NA NA NA ...
##  $ X.1          : logi  NA NA NA NA NA NA ...
##  $ X.2          : logi  NA NA NA NA NA NA ...
##  $ X.3          : logi  NA NA NA NA NA NA ...
##  $ X.4          : logi  NA NA NA NA NA NA ...
##  $ X.5          : logi  NA NA NA NA NA NA ...
```

```r
# Clean up Condition and Time.Point columns
colData$Condition <- trimws(as.character(colData$Condition))  # remove trailing spaces
colData$Condition <- factor(colData$Condition, levels = c("Mock", "SARS-CoV-2"))

# Convert Time.Point numeric → factor with "H" suffix
```

```r
colData$Time.Point <- paste0(colData$Time.Point, "H")
colData$Time.Point <- factor(colData$Time.Point, levels = c("24H", "72H"))

# Verify the structure again
str(colData)
```

```
## 'data.frame':    12 obs. of  10 variables:
##  $ Condition    : Factor w/ 2 levels "Mock","SARS-CoV-2": 1 1 1 2 2 2 1 1 1 2 ...
##  $ Sample       : int  1 2 3 1 2 3 1 2 3 1 ...
##  $ Time.Point   : Factor w/ 2 levels "24H","72H": 1 1 1 1 1 1 2 2 2 2 ...
##  $ SRA.Accession: Factor w/ 12 levels "SRR22269872",..: 12 11 10 9 8 7 6 5 4 3 ...
##  $ X            : logi  NA NA NA NA NA NA ...
##  $ X.1          : logi  NA NA NA NA NA NA ...
##  $ X.2          : logi  NA NA NA NA NA NA ...
##  $ X.3          : logi  NA NA NA NA NA NA ...
##  $ X.4          : logi  NA NA NA NA NA NA ...
##  $ X.5          : logi  NA NA NA NA NA NA ...
```

```r
table(colData$Condition)
```

```
##
##       Mock SARS-CoV-2
##          6          6
```

```r
table(colData$Time.Point)
```

```
##
## 24H 72H
##   6   6
```

```r
# Create DESeq dataset
dds <- DESeqDataSetFromMatrix(countData = round(countData),
                              colData = colData,
                              design = ~ Condition + Time.Point)
```

```
## converting counts to integer mode
```

```
##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]
```

```r
# DESeq normalization & modeling
dds <- DESeq(dds)
```

```
## estimating size factors
##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]
```

```
## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## -- note: fitType='parametric', but the dispersion trend was not well captured by the
##    function: y = a/x + b, and a local regression fit was automatically substituted.
##    specify fitType='local' or 'mean' to avoid this message next time.

##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]

## final dispersion estimates

## fitting model and testing
```

```
resultsNames(dds)
```

```
## [1] "Intercept"                    "Condition_SARS.CoV.2_vs_Mock"
## [3] "Time.Point_72H_vs_24H"
```

```
# ---------------------------------------------------
# Contrast 1: SARS-CoV-2 vs Mock (infection effect)
# ---------------------------------------------------
res1 <- results(dds, name = "Condition_SARS.CoV.2_vs_Mock", alpha = 0.05)
res1 <- lfcShrink(dds, coef = "Condition_SARS.CoV.2_vs_Mock", 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.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
```

```
# Sort by adjusted p-value
res1 <- res1[order(res1$padj), ]

# Filter for significant DEGs (padj < 0.05)
sig_res1 <- subset(res1, padj < 0.05)

# Save results
write.csv(as.data.frame(res1), "DEG_Mock_vs_SARS-CoV-2_all.csv", row.names = TRUE)
write.csv(as.data.frame(sig_res1), "DEG_Mock_vs_SARS-CoV-2_sig_padj0.05.csv", row.names = TRUE)

# ---------------------------------------------------
# Contrast 2: SARS-CoV-2 24H vs 72H (time effect)
# ---------------------------------------------------
if ("Time.Point_72H_vs_24H" %in% resultsNames(dds)) {
  res2 <- results(dds, name = "Time.Point_72H_vs_24H", alpha = 0.05)
  res2 <- lfcShrink(dds, coef = "Time.Point_72H_vs_24H", type = "apeglm")
} else {
  res2 <- results(dds, contrast = c("Time.Point", "72H", "24H"), alpha = 0.05)
}
```

```
## 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.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
```

```
# Sort and filter
res2 <- res2[order(res2$padj), ]
sig_res2 <- subset(res2, padj < 0.05)

# Save results
write.csv(as.data.frame(res2), "DEG_SARS-CoV-2_24h_vs_72h_all.csv", row.names = TRUE)
write.csv(as.data.frame(sig_res2), "DEG_SARS-CoV-2_24h_vs_72h_sig_padj0.05.csv", row.names = TRUE)

# ------------------------------------------------------
# Summary outputs
# ------------------------------------------------------
cat("Summary: Mock vs SARS-CoV-2\n")
```

```
## Summary: Mock vs SARS-CoV-2
```

```
summary(res1)
```

```
##
## out of 20143 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 40, 0.2%
## LFC < 0 (down)     : 25, 0.12%
## outliers [1]       : 2, 0.0099%
## low counts [2]     : 16584, 82%
## (mean count < 3)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
cat("\nNumber of significant DEGs (padj < 0.05):", nrow(sig_res1), "\n\n")
```

```
##
## Number of significant DEGs (padj < 0.05): 47
```

```
cat("Summary: 24H vs 72H\n")
```

```
## Summary: 24H vs 72H
```

```
summary(res2)
```

```
##
## out of 20143 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 114, 0.57%
## LFC < 0 (down)     : 137, 0.68%
## outliers [1]       : 2, 0.0099%
```

```
## low counts [2]     : 15548, 77%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
cat("\nNumber of significant DEGs (padj < 0.05):", nrow(sig_res2), "\n")
```

```
##
## Number of significant DEGs (padj < 0.05): 139
```

```
# -------------------------------
# 1. Prepare significant gene lists
# -------------------------------

sig_res1 <- res1[which(res1$padj < 0.05 & !is.na(res1$padj)), ]
sig_res2 <- res2[which(res2$padj < 0.05 & !is.na(res2$padj)), ]

genes1 <- gsub("\\..*", "", rownames(sig_res1))  # remove version numbers
genes2 <- gsub("\\..*", "", rownames(sig_res2))

cat("Significant DEGs:\n")
```

```
## Significant DEGs:
```

```
cat("Mock vs SARS-CoV-2:", length(genes1), "\n")
```

```
## Mock vs SARS-CoV-2: 47
```

```
cat("24H vs 72H:", length(genes2), "\n")
```

```
## 24H vs 72H: 139
```

```
# -------------------------------
# 2. Map Ensembl → Entrez IDs
# -------------------------------

genes1_map <- bitr(genes1, fromType = "ENSEMBL", toType = "ENTREZID", OrgDb = org.Hs.eg.db)
```

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

```
## Warning in bitr(genes1, fromType = "ENSEMBL", toType = "ENTREZID", OrgDb =
## org.Hs.eg.db): 12.77% of input gene IDs are fail to map...
```

```
genes2_map <- bitr(genes2, fromType = "ENSEMBL", toType = "ENTREZID", OrgDb = org.Hs.eg.db)
```

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

```
## Warning in bitr(genes2, fromType = "ENSEMBL", toType = "ENTREZID", OrgDb =
## org.Hs.eg.db): 8.63% of input gene IDs are fail to map...
```

```r
cat("\nMapped successfully:\n")
```

```
##
## Mapped successfully:
```

```r
cat("Contrast 1:", nrow(genes1_map), "mapped out of", length(genes1), "\n")
```

```
## Contrast 1: 41 mapped out of 47
```

```r
cat("Contrast 2:", nrow(genes2_map), "mapped out of", length(genes2), "\n")
```

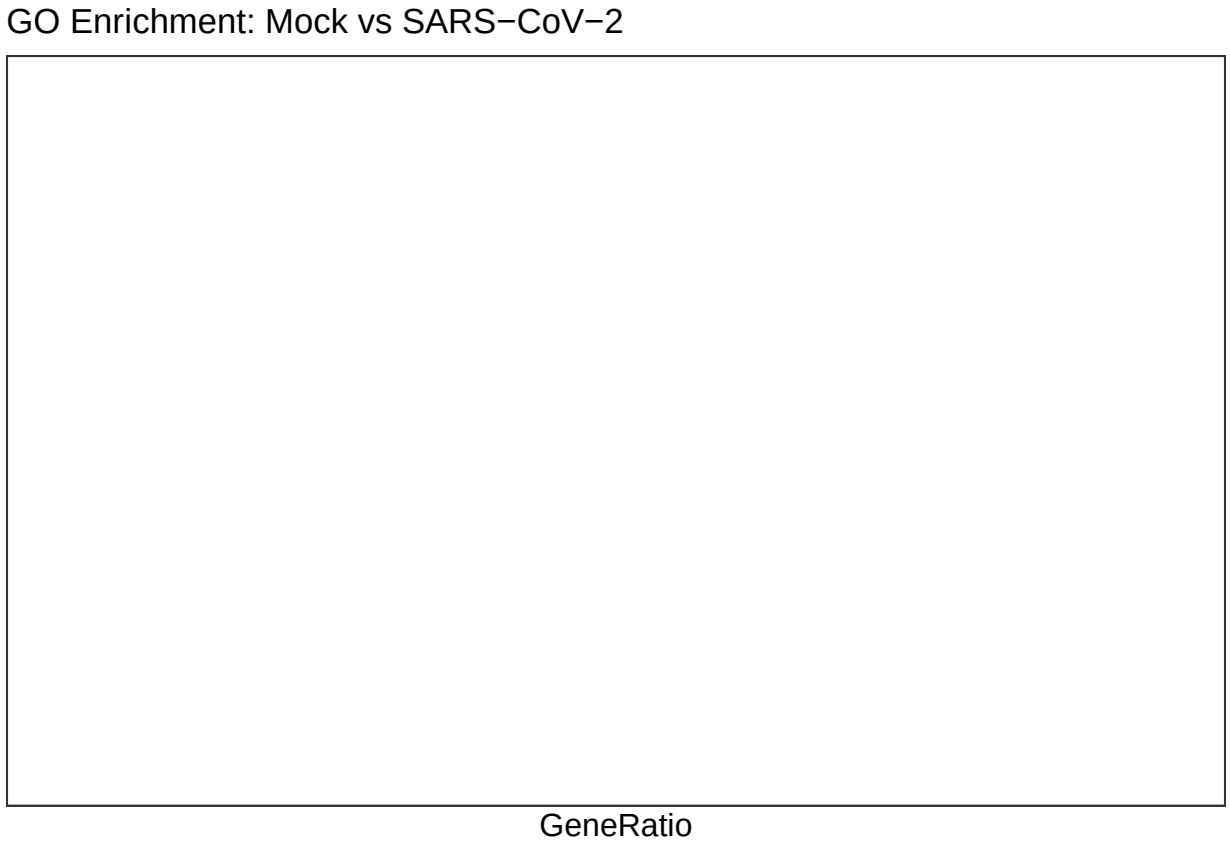
```
## Contrast 2: 128 mapped out of 139
```

```r
# --------------------------------
# 3. Run GO enrichment (Biological Process)
# --------------------------------

ego1 <- enrichGO(
  gene          = genes1_map$ENTREZID,
  OrgDb         = org.Hs.eg.db,
  keyType       = "ENTREZID",
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff  = 0.05,
  readable      = TRUE
)

ego2 <- enrichGO(
  gene          = genes2_map$ENTREZID,
  OrgDb         = org.Hs.eg.db,
  keyType       = "ENTREZID",
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff  = 0.05,
  readable      = TRUE
)

# --------------------------------
# 4. Save results and visualize
# --------------------------------

write.csv(as.data.frame(ego1), "GO_Mock_vs_SARS-CoV-2.csv", row.names = FALSE)
write.csv(as.data.frame(ego2), "GO_24H_vs_72H.csv", row.names = FALSE)

# Top categories
library(enrichplot)
```

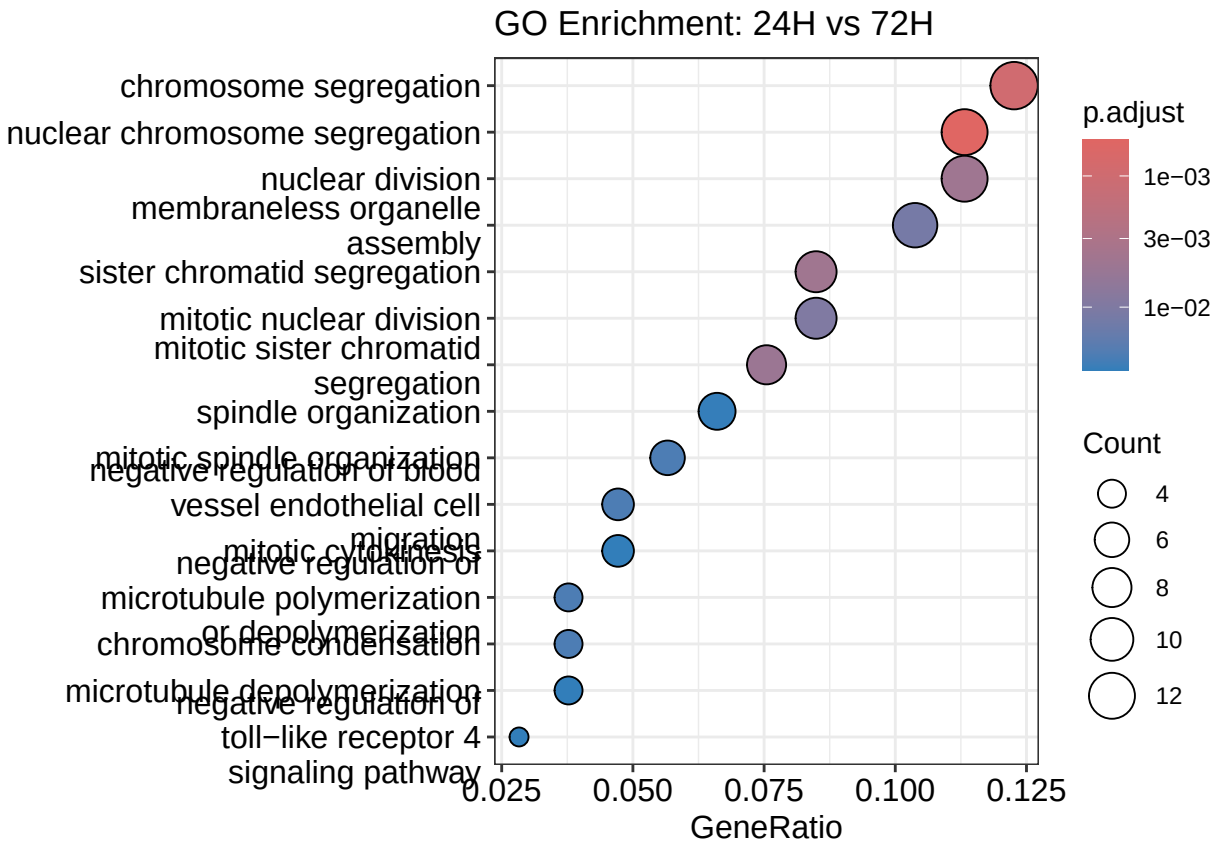
```
## enrichplot v1.30.1 Learn more at https://yulab-smu.top/contribution-knowledge-mining/
##
## Please cite:
##
```

```
## S Xu, E Hu, Y Cai, Z Xie, X Luo, L Zhan, W Tang, Q Wang, B Liu, R Wang,
## W Xie, T Wu, L Xie, G Yu. Using clusterProfiler to characterize
## multiomics data. Nature Protocols. 2024, 19(11):3292-3320
```

```
dotplot(ego1, showCategory = 15, title = "GO Enrichment: Mock vs SARS-CoV-2")
```




```
dotplot(ego2, showCategory = 15, title = "GO Enrichment: 24H vs 72H")
```


GO Enrichment: 24H vs 72H
chromosome segregation
nuclear chromosome segregation
nuclear division
membraneless organelle assembly
sister chromatid segregation
mitotic nuclear division
mitotic sister chromatid segregation
spindle organization
mitotic spindle organization
negative regulation of blood vessel endothelial cell migration
mitotic cytokinesis
negative regulation of microtubule polymerization or depolymerization
chromosome condensation
microtubule depolymerization
negative regulation of toll−like receptor 4 signaling pathway
0.025
0.050
0.075
0.100
0.125
GeneRatio
p.adjust
1e−03
3e−03
1e−02
Count
4
6
8
10
12