

Visualizing Bile Acids Proteomics results

```
library(clusterProfiler)
library(EnhancedVolcano)
library(org.Hs.eg.db)
library(tidyverse)
library(readxl)

mq <- read_tsv(here::here("normalyzer_limma_de_results_annotated.tsv"))

## Rows: 5705 Columns: 35
## -- Column specification -----
## Delimiter: "\t"
## chr (4): Majority protein IDs, Protein IDs, Protein names, Gene names
## dbl (31): Cc-Lc_PValue, Cm-Lm_PValue, Cc-Lc_AdjPVal, Cm-Lm_AdjPVal, Cc-Lc_lo...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

meta <- read_tsv(here::here("norm_meta.tsv"))

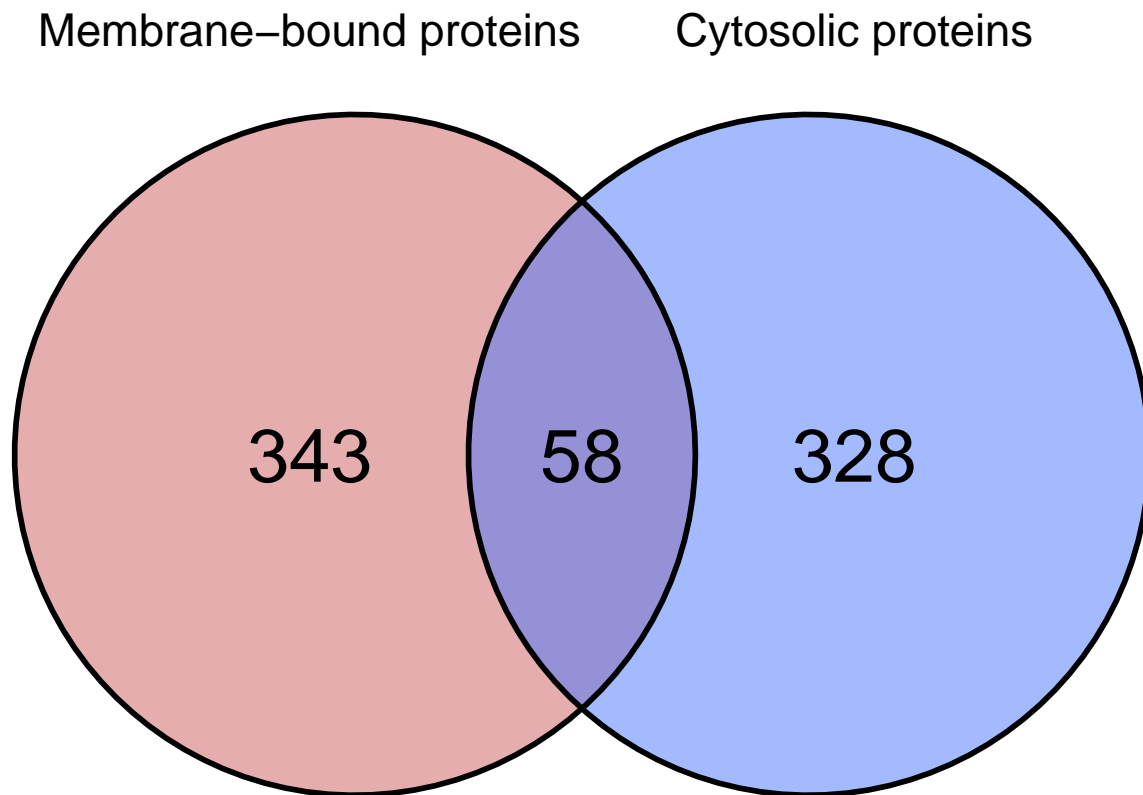
## Rows: 24 Columns: 2
## -- Column specification -----
## Delimiter: "\t"
## chr (2): sample, group
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

mq_res <- mq %>%
  select(contains("Log"), contains("Adj"), contains("Majority"), contains("Gene"))
mq_mat <- mq %>%
  select(`Gene names`, `Majority protein IDs`, matches("(C|L)[1-6](c|m)"))
#Convert the results table into a list and unify the columns names.
#Invert the foldChanges, so that comparisons are L-C instead of C-L.
#Reduce Gene names to just one gene for analysis
mq_res_list <- list(select(mq_res, -contains("Cc-Lc")),
  select(mq_res, -contains("Cm-Lm"))) %>%
  set_names(nm = c("Membrane", "Cyto")) %>%
  map(rename_with, ~str_remove(., "C(c|m)-L(c|m)_"), matches("C(c|m)-L(c|m)_")) %>%
  map(mutate, across(contains("log2"), ~ -.x)) %>%
  map(separate, `Gene names`, into = letters[1:2], extra = "drop", sep = ";") %>%
  map(mutate, Genes = case_when(
    str_detect(a, "orf") & !is.na(b) & str_detect(b, "orf", negate = T) ~ b,
    TRUE ~ a
  ), Gene_rows = Genes)
```

```
## Warning: Expected 2 pieces. Missing pieces filled with 'NA' in 5473 rows [3, 4, 6, 7, 8, 10, 11, 12,
## Expected 2 pieces. Missing pieces filled with 'NA' in 5473 rows [3, 4, 6, 7, 8, 10, 11, 12, 13, 14, 15]
```

Venn diagram

```
#Filter results to only include DA proteins
da_list <- mq_res_list %>%
  map(filter, AdjPVal < 0.05, abs(log2FoldChange) > 1)
#Pull just the Protein IDs for DA proteins
da_genes <- da_list %>% map(pull, `Majority protein IDs`) %>%
  set_names(nm = c("Membrane-bound proteins", "Cytosolic proteins"))
ggvenn::ggvenn(da_genes, text_size = 10, set_name_size = 6,
  fill_color = c("indianred", "royalblue1"), show_percentage = F)
```



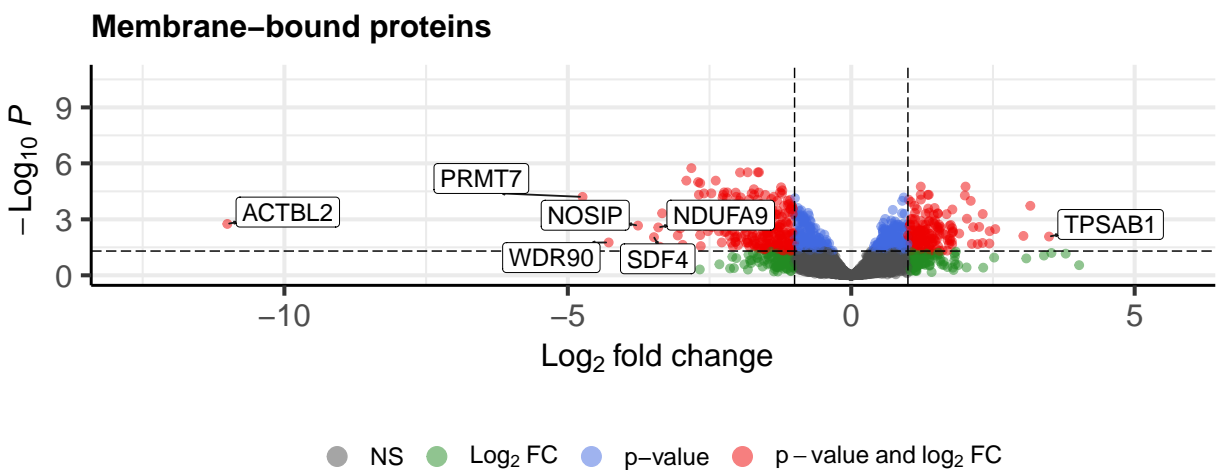
```
#ggsave(here::here("Venn diagram DA proteins.png"), units = "in", dpi = 720, width = 8, height = 6)
```

Get identities of common DE proteins

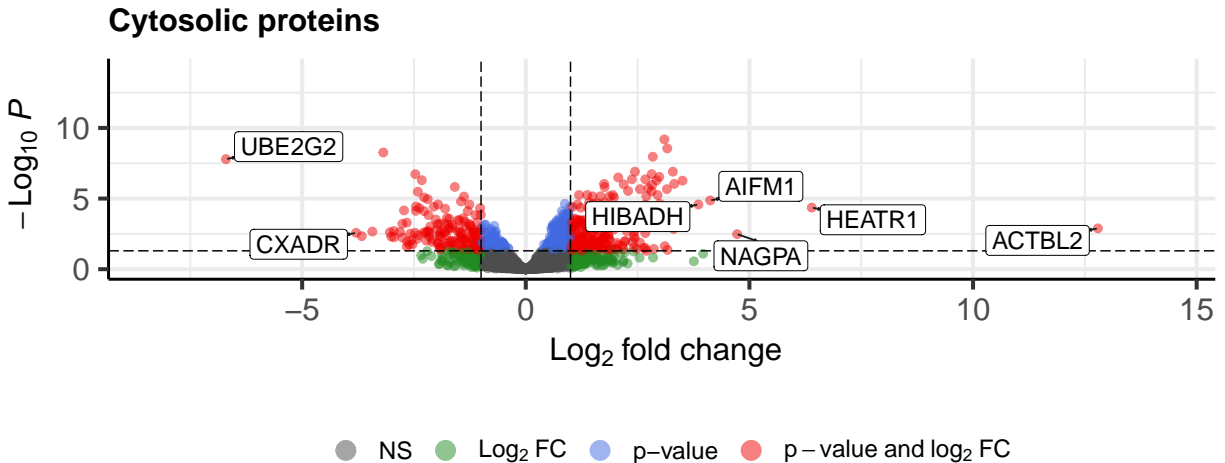
```
common_top <- da_genes %>%
  reduce(intersect)
mq_intersect_top <- mq_res %>%
  filter(`Majority protein IDs` %in% common_top)
#write_csv(mq_intersect_top, here::here("Shared DE proteins.csv"))
```

Volcano plots

```
df <- mq_res_list[[1]]
rows <- map(mq_res_list, pull, var = Gene_rows)
titles <- names(da_genes)
top_hits <- mq_res_list %>%
  map(filter, AdjPVal < 0.05) %>%
  map(arrange, desc(abs(log2FoldChange))) %>%
  map(slice, 1:7) %>%
  map(pull, var = Gene_rows)
plot_list <- list(df = mq_res_list, rows = rows, title = titles, selectLab = top_hits)
#Create a custom volcano plot function
customvolcano <- function(df, rows, title, selectLab="") {
  print(EnhancedVolcano(toptable = df, lab = rows, x = "log2FoldChange", y = "AdjPVal",
    pCutoff = 0.05, FCcutoff = 1, legendPosition = 'bottom',
    title = title, subtitle = NULL,
    selectLab = selectLab, drawConnectors = T, boxedLabels = T))
}
#pdf(here::here("Volcano plots top 7 hits labelled.pdf"), width = 10)
pwalk(plot_list, customvolcano)
```



total = 5705 variables



total = 5705 variables

```
#dev.off()
```

Functional enrichment

```
set.seed(711)
#Get the whole human genome as universe
human_genom <- org.Hs.egSYMBOL
mapped_genes <- mappedRkeys(human_genom)
h <- read.gmt(list.files(path = here::here("."), pattern = "h.all"))
c2_cp <- read.gmt(list.files(path = here::here("."), pattern = "c2.cp"))
c2_noID <- c2_cp %>%
  mutate(term = str_remove(term, "[:alpha:]+_"))
#Set up lists of genes for analysis
#Pull just the gene names for DA proteins
da_symbols <- da_list %>% map(pull, `Genes`) %>%
  set_names(nm = c("Membrane-bound proteins", "Cytosolic proteins"))
enrich_h_list <- map(da_symbols, ~enricher(.,
  TERM2GENE = h,
  universe = mapped_genes,
  pAdjustMethod = "BH",
  pvalueCutoff = 0.05,
  qvalueCutoff = 0.05))
enrich_c2_list <- map(da_symbols, ~enricher(.,
  TERM2GENE = c2_noID,
  universe = mapped_genes,
  pAdjustMethod = "BH",
  pvalueCutoff = 0.05,
  qvalueCutoff = 0.05))
ego_bp <- map(da_symbols, ~enrichGO(gene = .,
  OrgDb = org.Hs.eg.db,
  universe = mapped_genes,
  keyType = 'SYMBOL',
```

```

        ont           = "BP",
        pAdjustMethod = "BH",
        pvalueCutoff  = 0.05,
        qvalueCutoff  = 0.05))
ego_bp <- map(ego_bp, enrichplot::pairwise_termsim)
ego_bp2 <- ego_bp %>%
  map(~clusterProfiler::simplify(., cutoff=0.4, by="p.adjust", select_fun=min))

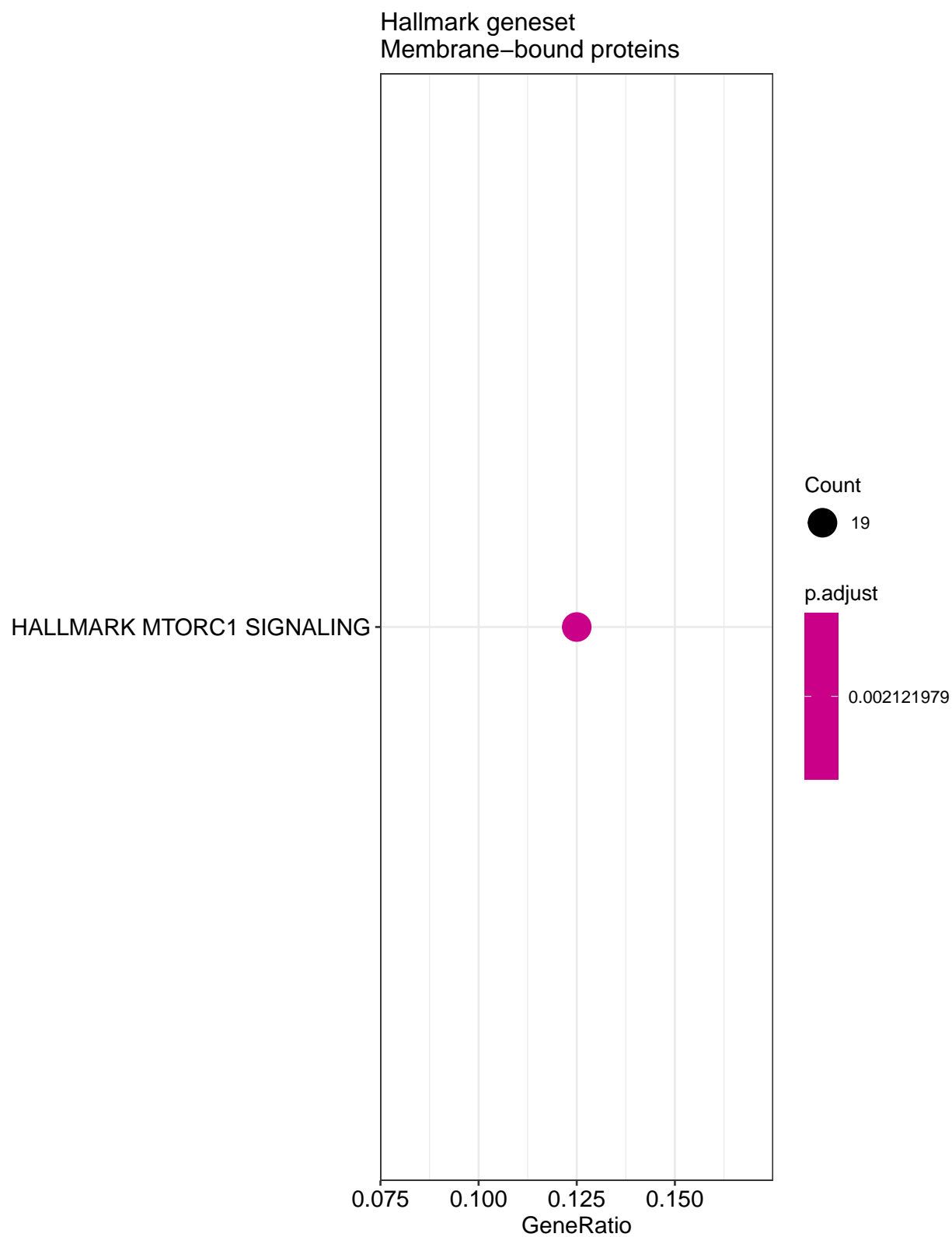
```

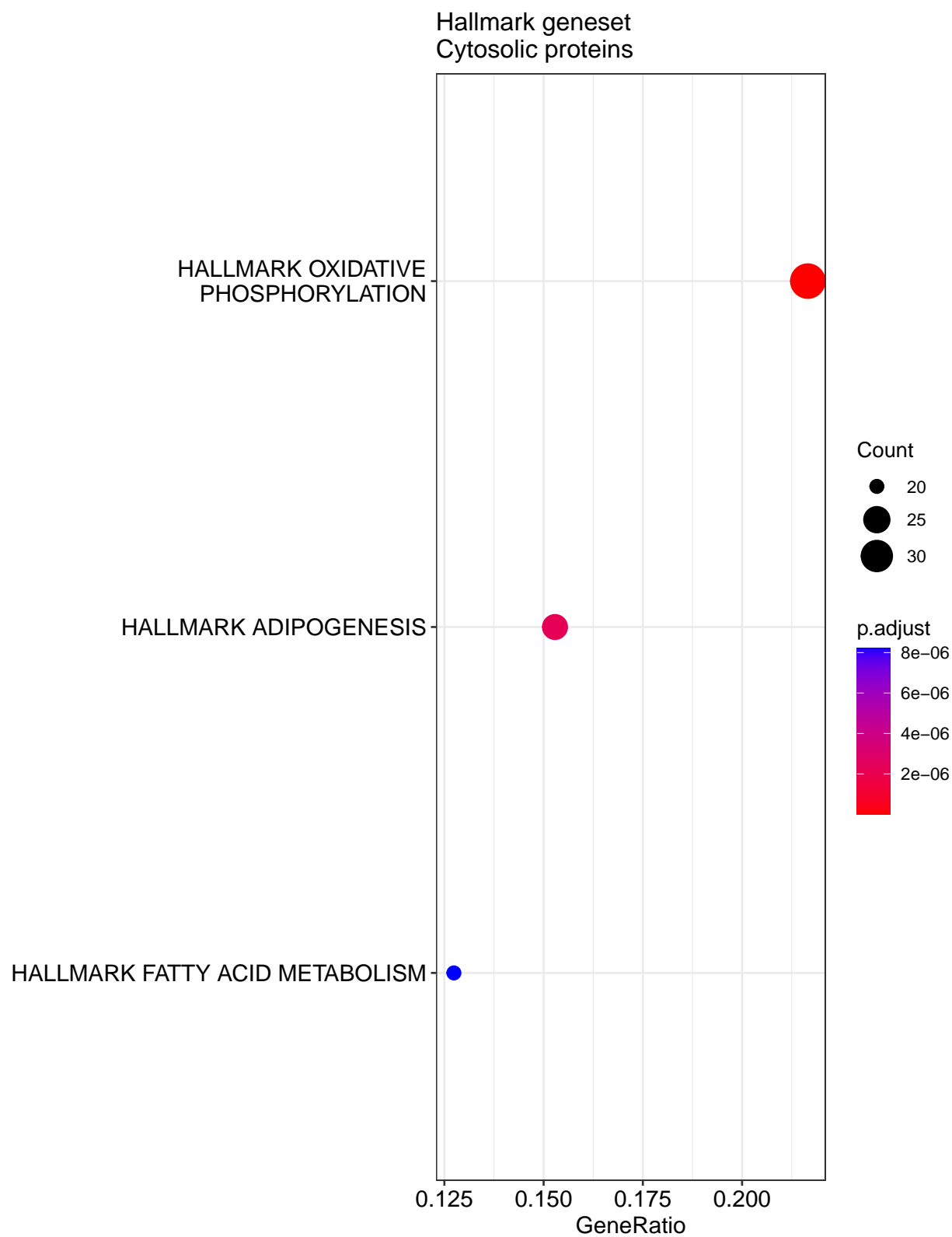
Plot the functional enrichment results

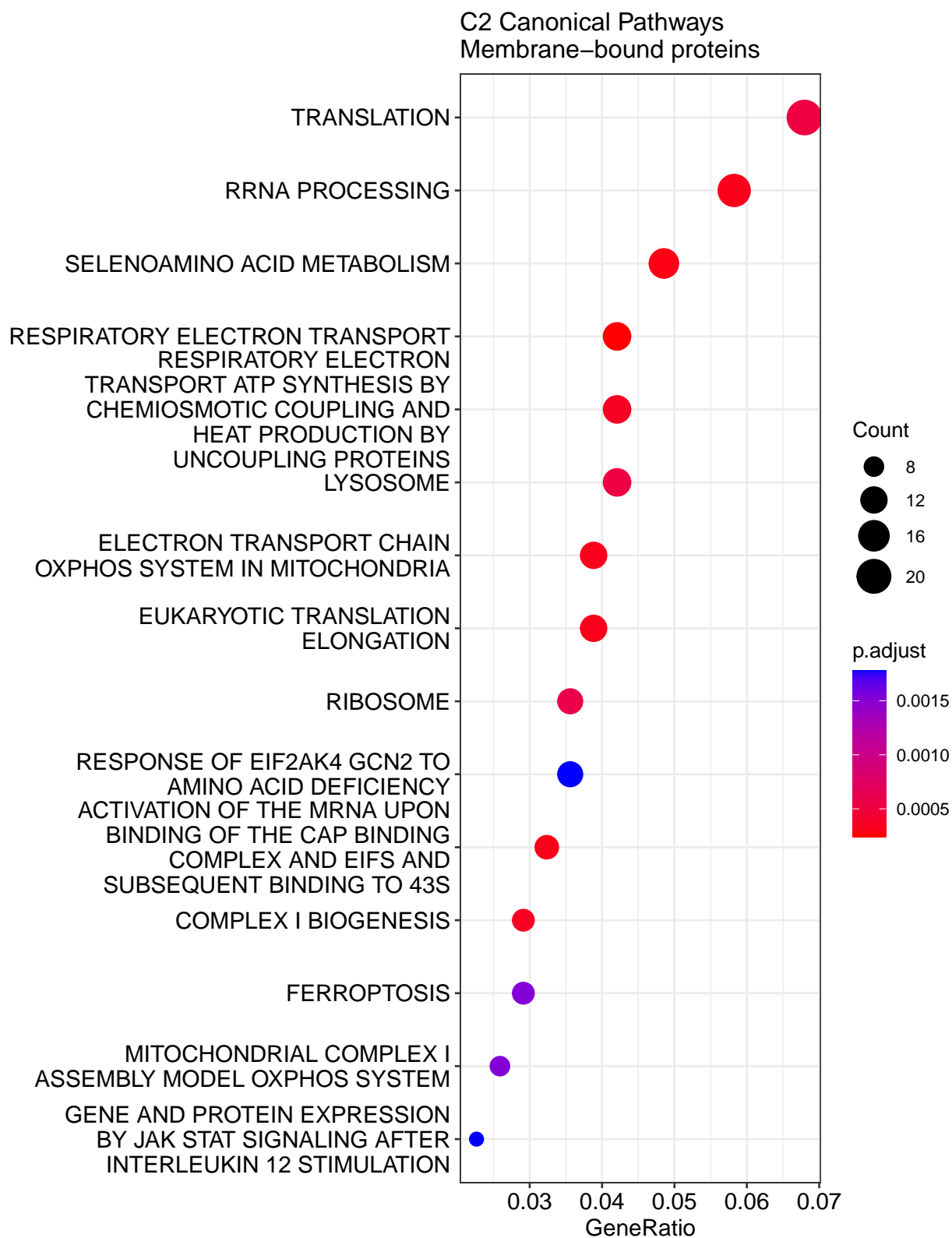
```

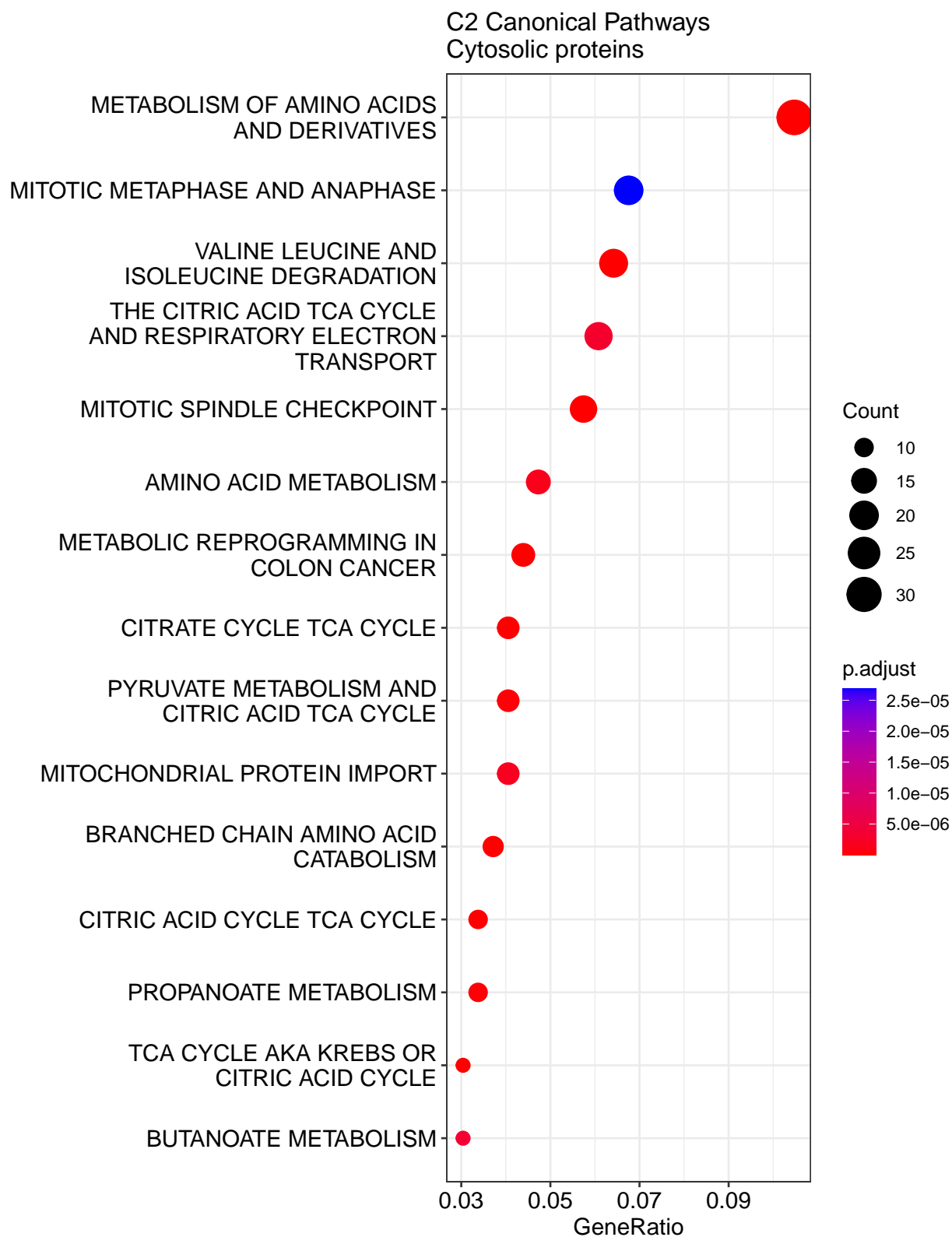
dot_titles <- rep(c("Hallmark geneset", "C2 Canonical Pathways", "GO BP"), each = 2) %>%
  str_c(names(da_genes), sep = "\n")
#pdf(here::here("Functional enrichment analyses.pdf"), height = 9, width = 7)
walk2(c(enrich_h_list, enrich_c2_list, ego_bp2), dot_titles,
  ~print(dotplot(.x, showCategory = 15) + ggtitle(.y)))

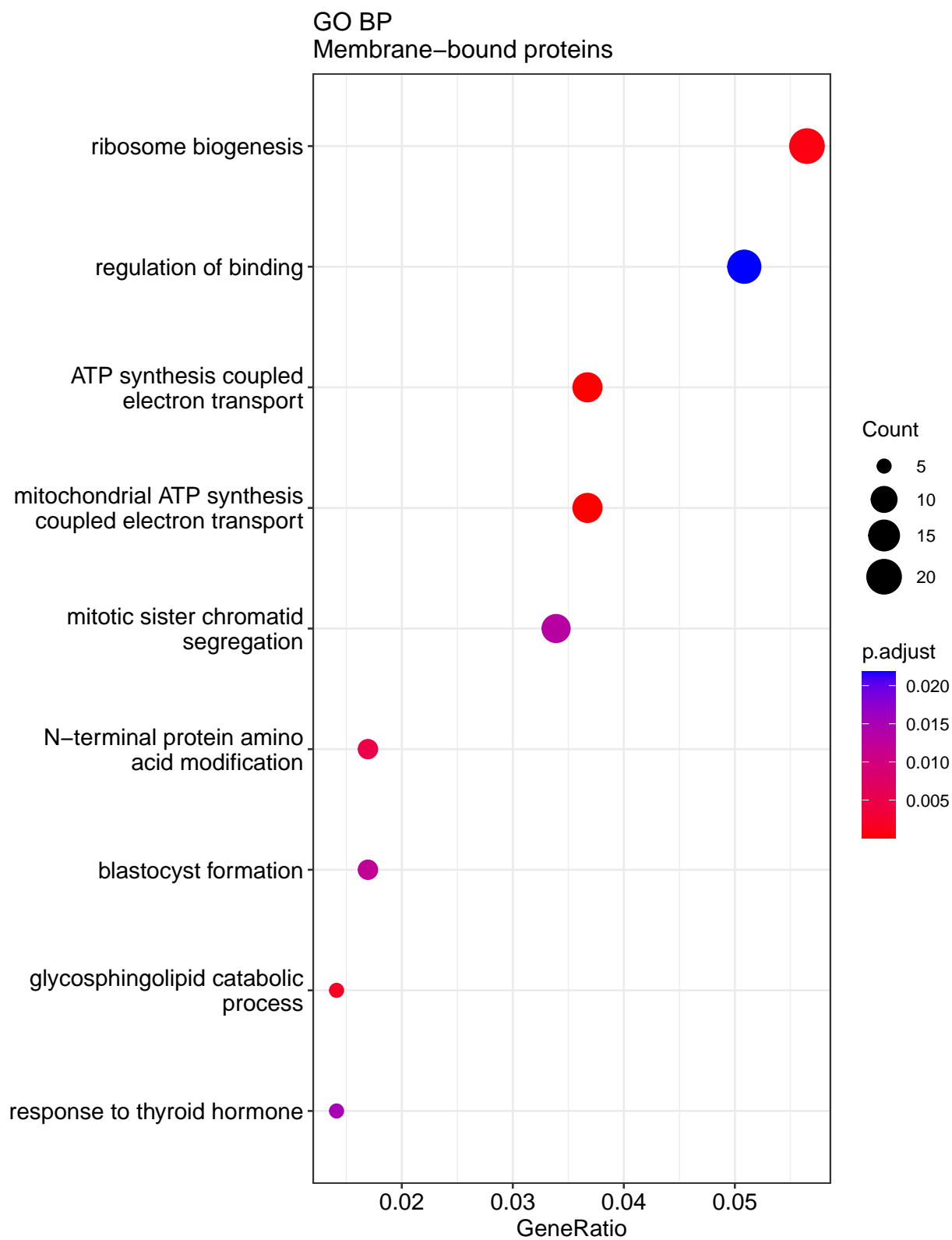
```

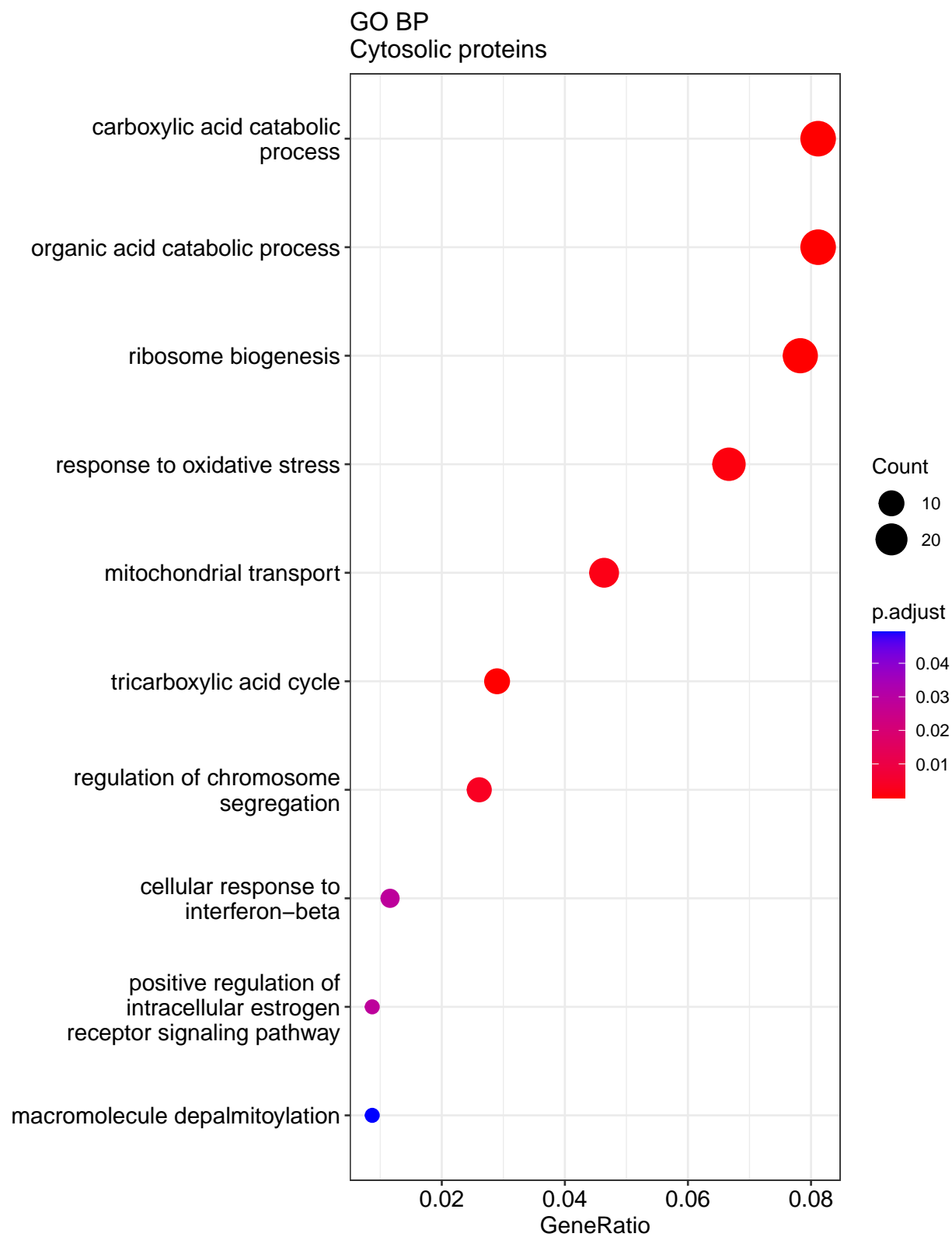












```
#dev.off()
```

Session Info

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Canada.1252 LC_CTYPE=English_Canada.1252
## [3] LC_MONETARY=English_Canada.1252 LC_NUMERIC=C
## [5] LC_TIME=English_Canada.1252
##
## attached base packages:
## [1] stats4      stats      graphics  grDevices  utils      datasets  methods
## [8] base
##
## other attached packages:
## [1] readxl_1.4.0      forcats_0.5.1      stringr_1.4.0
## [4] dplyr_1.0.9       purrr_0.3.4        readr_2.1.2
## [7] tidyr_1.2.0       tibble_3.1.7       tidyverse_1.3.1
## [10] org.Hs.eg.db_3.14.0 AnnotationDbi_1.56.2 IRanges_2.28.0
## [13] S4Vectors_0.32.4 Biobase_2.54.0      BiocGenerics_0.40.0
## [16] EnhancedVolcano_1.12.0 ggrepel_0.9.1       ggplot2_3.3.6
## [19] clusterProfiler_4.2.2
##
## loaded via a namespace (and not attached):
## [1] shadowtext_0.1.2      backports_1.4.1      fastmatch_1.1-3
## [4] plyr_1.8.7           igraph_1.3.2         lazyeval_0.2.2
## [7] splines_4.1.2        BiocParallel_1.28.3  GenomeInfoDb_1.30.1
## [10] digest_0.6.29        yulab.utils_0.0.5    htmltools_0.5.2
## [13] GOSeqSim_2.20.0       viridis_0.6.2        GO.db_3.14.0
## [16] fansi_1.0.2          magrittr_2.0.3       memoise_2.0.1
## [19] tzdb_0.3.0           Biostrings_2.62.0    graphlayouts_0.8.0
## [22] modelr_0.1.8         extrafont_0.18       vroom_1.5.7
## [25] extrafontdb_1.0      enrichplot_1.14.2    colorspace_2.0-3
## [28] rvest_1.0.2          blob_1.2.3           haven_2.4.3
## [31] xfun_0.29            crayon_1.5.1         RCurl_1.98-1.7
## [34] jsonlite_1.8.0       scatterpie_0.1.7     ape_5.6-2
## [37] glue_1.6.2           polyclip_1.10-0      gtable_0.3.0
## [40] zlibbioc_1.40.0      XVector_0.34.0       proj4_1.0-11
## [43] Rttf2pt1_1.3.10      maps_3.4.0           scales_1.2.0
## [46] DOSE_3.20.1          DBI_1.1.3            Rcpp_1.0.8.3
## [49] viridisLite_0.4.0    gridGraphics_0.5-1   tidytree_0.3.9
## [52] bit_4.0.4            httr_1.4.3           fgsea_1.20.0
## [55] RColorBrewer_1.1-3   ellipsis_0.3.2       pkgconfig_2.0.3
```

## [58] farver_2.1.0	dbplyr_2.2.1	here_1.0.1
## [61] utf8_1.2.2	labeling_0.4.2	ggplotify_0.1.0
## [64] tidyselect_1.1.2	rlang_1.0.3	reshape2_1.4.4
## [67] cellranger_1.1.0	munsell_0.5.0	tools_4.1.2
## [70] cachem_1.0.6	downloader_0.4	cli_3.3.0
## [73] generics_0.1.3	RSQLite_2.2.14	broom_1.0.0
## [76] evaluate_0.15	fastmap_1.1.0	yaml_2.2.2
## [79] ggtree_3.2.1	fs_1.5.2	knitr_1.39
## [82] bit64_4.0.5	tidygraph_1.2.1	KEGGREST_1.34.0
## [85] ggraph_2.0.5	nlme_3.1-153	ash_1.0-15
## [88] ggrrastr_1.0.1	aplot_0.1.6	ggvenn_0.1.9
## [91] xml2_1.3.3	DO.db_2.9	compiler_4.1.2
## [94] rstudioapi_0.13	beeswarm_0.4.0	png_0.1-7
## [97] reprex_2.0.1	treeio_1.18.1	tweenr_1.0.2
## [100] stringi_1.7.6	highr_0.9	ggalt_0.4.0
## [103] lattice_0.20-45	Matrix_1.3-4	vcvtrs_0.4.1
## [106] pillar_1.8.0	lifecycle_1.0.1	data.table_1.14.2
## [109] bitops_1.0-7	patchwork_1.1.1	qvalue_2.26.0
## [112] R6_2.5.1	KernSmooth_2.23-20	gridExtra_2.3
## [115] vipor_0.4.5	MASS_7.3-54	assertthat_0.2.1
## [118] rprojroot_2.0.3	withr_2.5.0	GenomeInfoDbData_1.2.7
## [121] parallel_4.1.2	hms_1.1.1	grid_4.1.2
## [124] ggfun_0.0.6	rmarkdown_2.14	ggforce_0.3.3
## [127] lubridate_1.8.0	ggbeeswarm_0.6.0	