

Transcriptomics R file

Hang Lei and Sophia Madjarova

2025-05-03

Load Required Packages

```
required_cran <- c("readxl", "dplyr", "stringr", "tibble", "tidyr", "ggplot2", "pheatmap")
installed <- rownames(installed.packages())
to_install <- setdiff(required_cran, installed)
if(length(to_install)) install.packages(to_install)

if (!requireNamespace("BiocManager", quietly = TRUE)) install.packages("BiocManager")
if (!"limma" %in% installed) BiocManager::install("limma")

library(readxl)
library(dplyr)
library(stringr)
library(tidyr)
library(ggplot2)
library(tibble)
library(limma)
library(pheatmap)
```

Load and Filter Data

```
data <- read_excel("GSE285580_All_data.xlsx")
```

```
## New names:
## * ' ' -> '...3'
## * ' ' -> '...4'
## * ' ' -> '...5'
## * ' ' -> '...6'
## * ' ' -> '...7'
## * ' ' -> '...8'
## * ' ' -> '...9'
## * ' ' -> '...10'
## * ' ' -> '...11'
## * ' ' -> '...12'
## * ' ' -> '...13'
## * ' ' -> '...14'
## * ' ' -> '...15'
## * ' ' -> '...16'
```

```

## * '' -> '...17'
## * '' -> '...18'
## * '' -> '...19'
## * '' -> '...20'
## * '' -> '...21'
## * '' -> '...22'
## * '' -> '...23'
## * '' -> '...24'
## * '' -> '...25'
## * '' -> '...26'
## * '' -> '...27'
## * '' -> '...28'
## * '' -> '...29'
## * '' -> '...30'
## * '' -> '...31'
## * '' -> '...32'
## * '' -> '...33'
## * '' -> '...34'
## * '' -> '...35'
## * '' -> '...36'
## * '' -> '...37'
## * '' -> '...38'
## * '' -> '...39'
## * '' -> '...40'
## * '' -> '...41'
## * '' -> '...42'
## * '' -> '...43'
## * '' -> '...44'
## * '' -> '...45'
## * '' -> '...46'
## * '' -> '...47'
## * '' -> '...48'
## * '' -> '...49'
## * '' -> '...50'
## * '' -> '...51'
## * '' -> '...52'
## * '' -> '...53'
## * '' -> '...54'
## * '' -> '...55'
## * '' -> '...56'
## * '' -> '...57'
## * '' -> '...58'
## * '' -> '...59'
## * '' -> '...60'
## * '' -> '...61'
## * '' -> '...62'
## * '' -> '...63'
## * '' -> '...64'
## * '' -> '...65'
## * '' -> '...66'
## * '' -> '...67'
## * '' -> '...68'
## * '' -> '...69'
## * '' -> '...70'

```

```
## * '' -> '...71'
## * '' -> '...72'
## * '' -> '...73'
## * '' -> '...74'
## * '' -> '...75'
## * '' -> '...76'
## * '' -> '...77'
## * '' -> '...78'
## * '' -> '...79'
## * '' -> '...80'
## * '' -> '...81'
## * '' -> '...82'
## * '' -> '...83'
## * '' -> '...84'
## * '' -> '...85'
## * '' -> '...86'
## * '' -> '...87'
## * '' -> '...88'
## * '' -> '...89'
## * '' -> '...90'
## * '' -> '...91'
## * '' -> '...92'
## * '' -> '...93'
## * '' -> '...94'
## * '' -> '...95'
## * '' -> '...96'
## * '' -> '...97'
## * '' -> '...98'
## * '' -> '...99'
## * '' -> '...100'
## * '' -> '...101'
## * '' -> '...102'
```

```
new_data <- read_excel("GSE285580_All_data.xlsx", skip = 48)
new_data_unique <- new_data %>% filter(!duplicated(miR_name))
```

Extract Expression Matrix

```
expr_cols <- grep("\\(norm\\)$", colnames(new_data), value = TRUE)
expr_matrix <- new_data_unique %>%
  dplyr::select(miR_name, all_of(expr_cols)) %>%
  column_to_rownames("miR_name")
colnames(expr_matrix) <- gsub("\\(norm\\)", "", colnames(expr_matrix))
```

Create Sample Metadata

```
sample_info <- data.frame(
  sample_id = colnames(expr_matrix),
  genotype = c(rep("HbAA", 13), rep("HbAS", 13), rep("HbSS", 13))
```

```
)
all(sample_info$sample_id == colnames(expr_matrix))
```

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

Differential Expression Analysis

```
expr_matrix <- as.matrix(expr_matrix)
group <- factor(sample_info$genotype, levels = c("HbAA", "HbAS", "HbSS"))
design <- model.matrix(~ 0 + group)
colnames(design) <- levels(group)
fit <- lmFit(expr_matrix, design)

contrast.matrix <- makeContrasts(
  HbASvsHbAA = HbAS - HbAA,
  HbSSvsHbAA = HbSS - HbAA,
  HbSSvsHbAS = HbSS - HbAS,
  levels = design
)

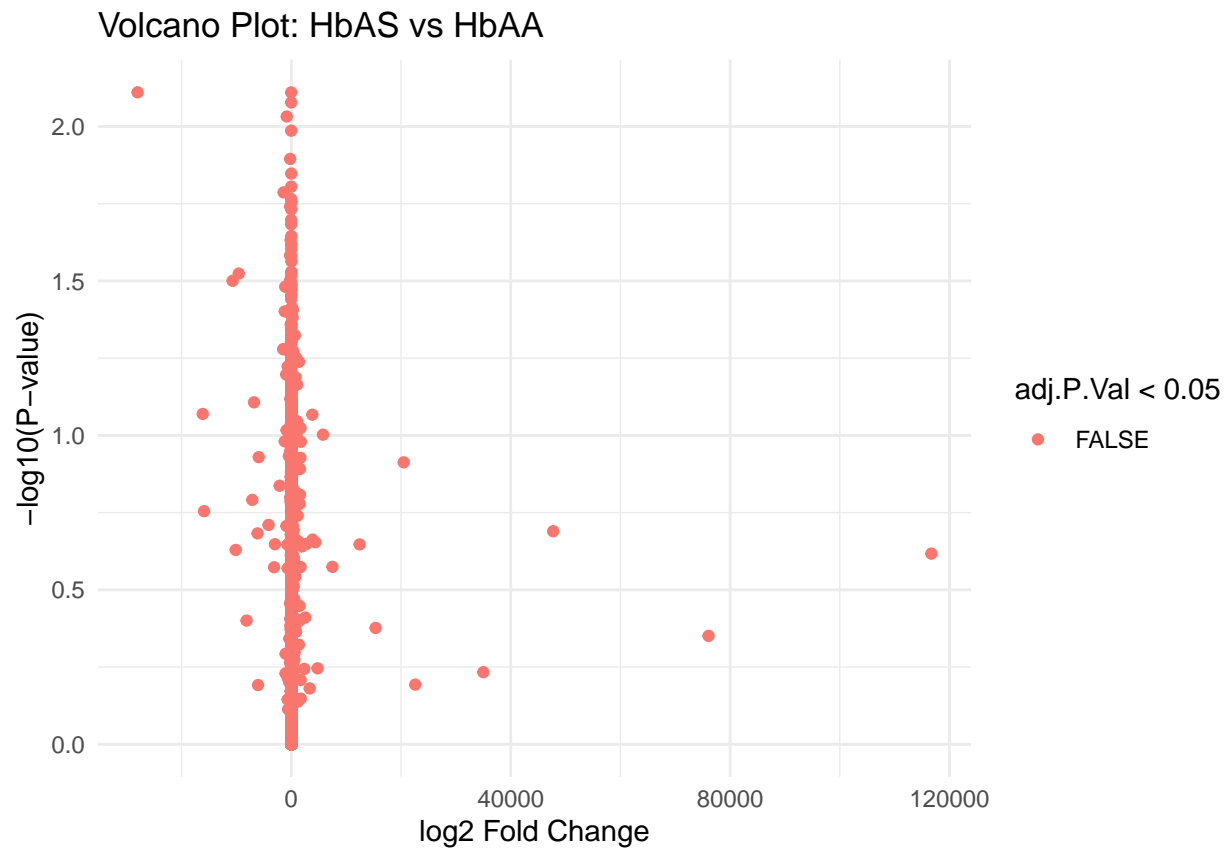
fit2 <- contrasts.fit(fit, contrast.matrix)
fit2 <- eBayes(fit2)
```

Extract DE Results

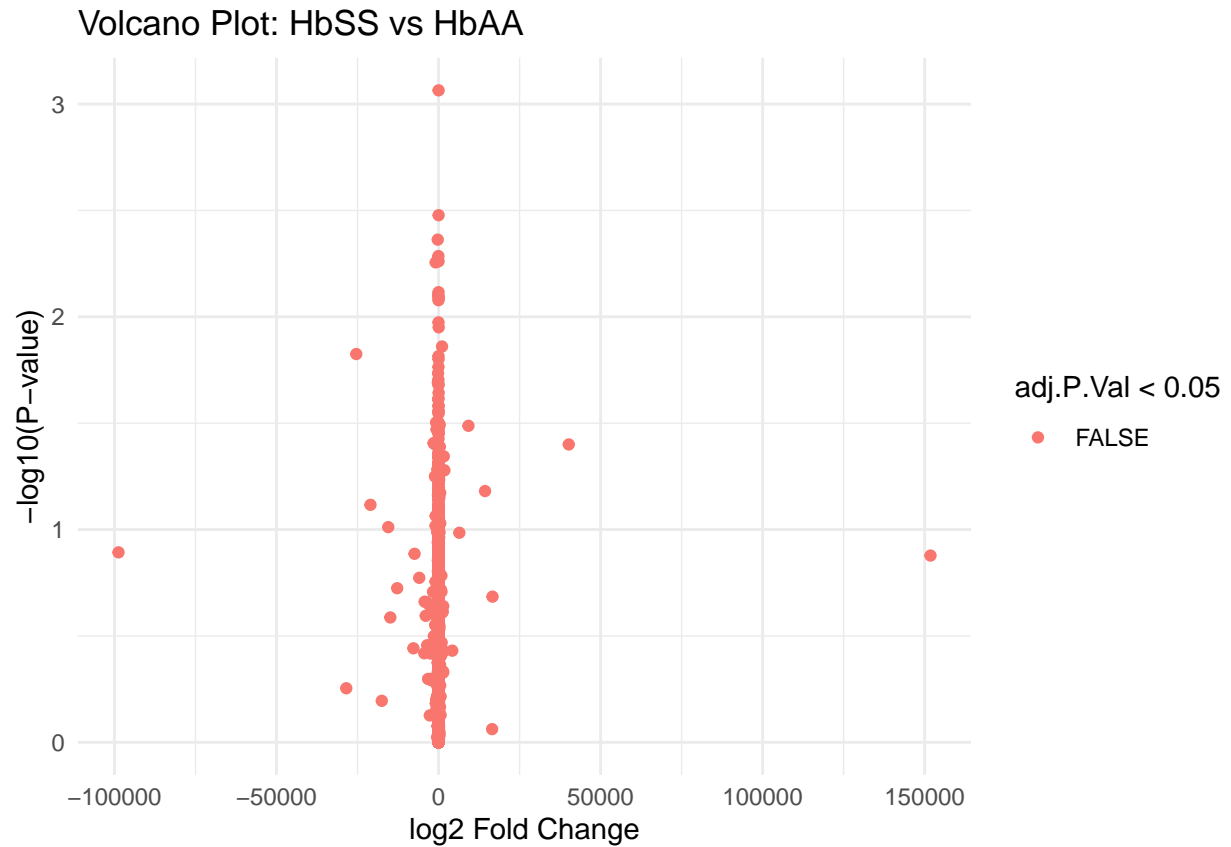
```
results_AS <- topTable(fit2, coef = "HbASvsHbAA", adjust = "fdr", number = Inf)
results_SS <- topTable(fit2, coef = "HbSSvsHbAA", adjust = "fdr", number = Inf)
results_SSvsAS <- topTable(fit2, coef = "HbSSvsHbAS", adjust = "fdr", number = Inf)
```

Volcano Plots

```
ggplot(results_AS, aes(x = logFC, y = -log10(P.Value), color = adj.P.Val < 0.05)) +
  geom_point() + theme_minimal() +
  labs(title = "Volcano Plot: HbAS vs HbAA", x = "log2 Fold Change", y = "-log10(P-value)")
```

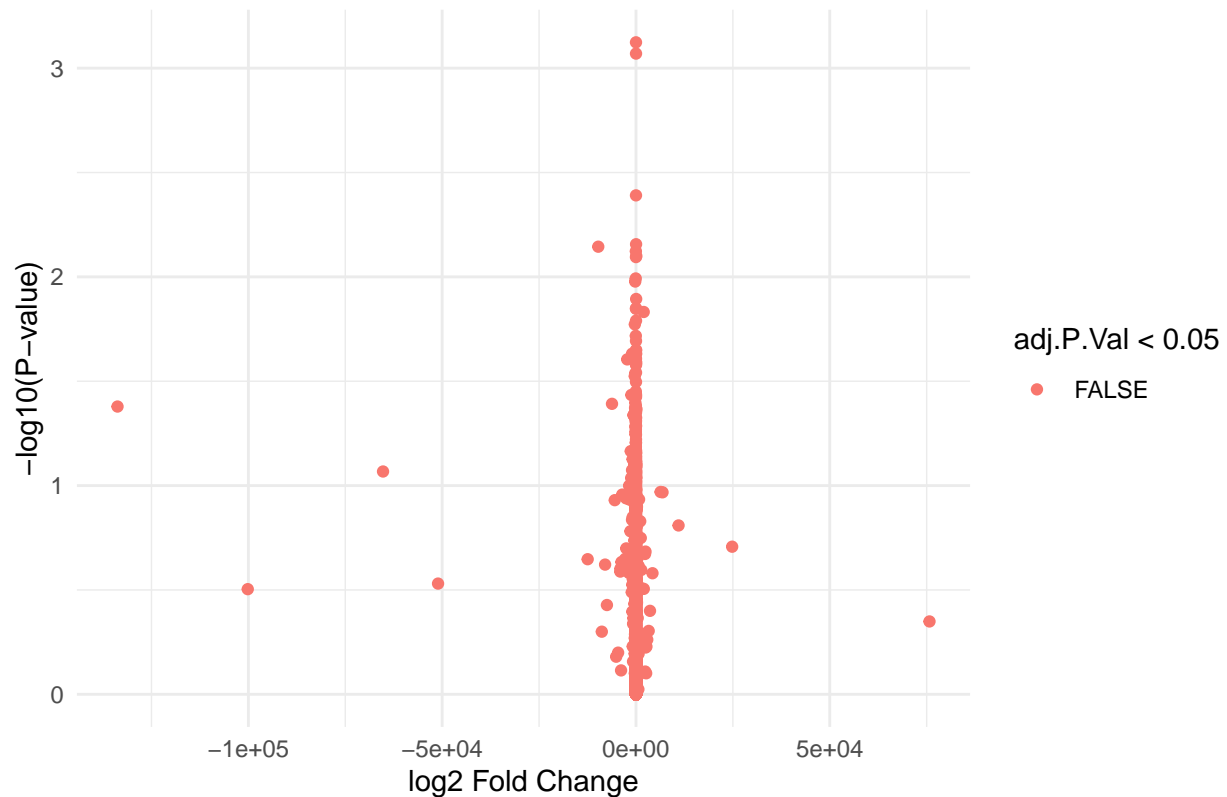


```
ggplot(results_SS, aes(x = logFC, y = -log10(P.Value), color = adj.P.Val < 0.05)) +
  geom_point() + theme_minimal() +
  labs(title = "Volcano Plot: HbSS vs HbAA", x = "log2 Fold Change", y = "-log10(P-value)")
```



```
ggplot(results_SSvsAS, aes(x = logFC, y = -log10(P.Value), color = adj.P.Val < 0.05)) +  
  geom_point() + theme_minimal() +  
  labs(title = "Volcano Plot: HbSS vs HbAS", x = "log2 Fold Change", y = "-log10(P-value)")
```

Volcano Plot: HbSS vs HbAS



Identify Significant miRNAs

```
sig_AS <- results_AS %>% filter(adj.P.Val < 0.05)
sig_SS <- results_SS %>% filter(adj.P.Val < 0.05)
sig_SSvsAS <- results_SSvsAS %>% filter(adj.P.Val < 0.05)
```

Heatmap of Top 20 DE miRNAs (HbSS vs HbAA)

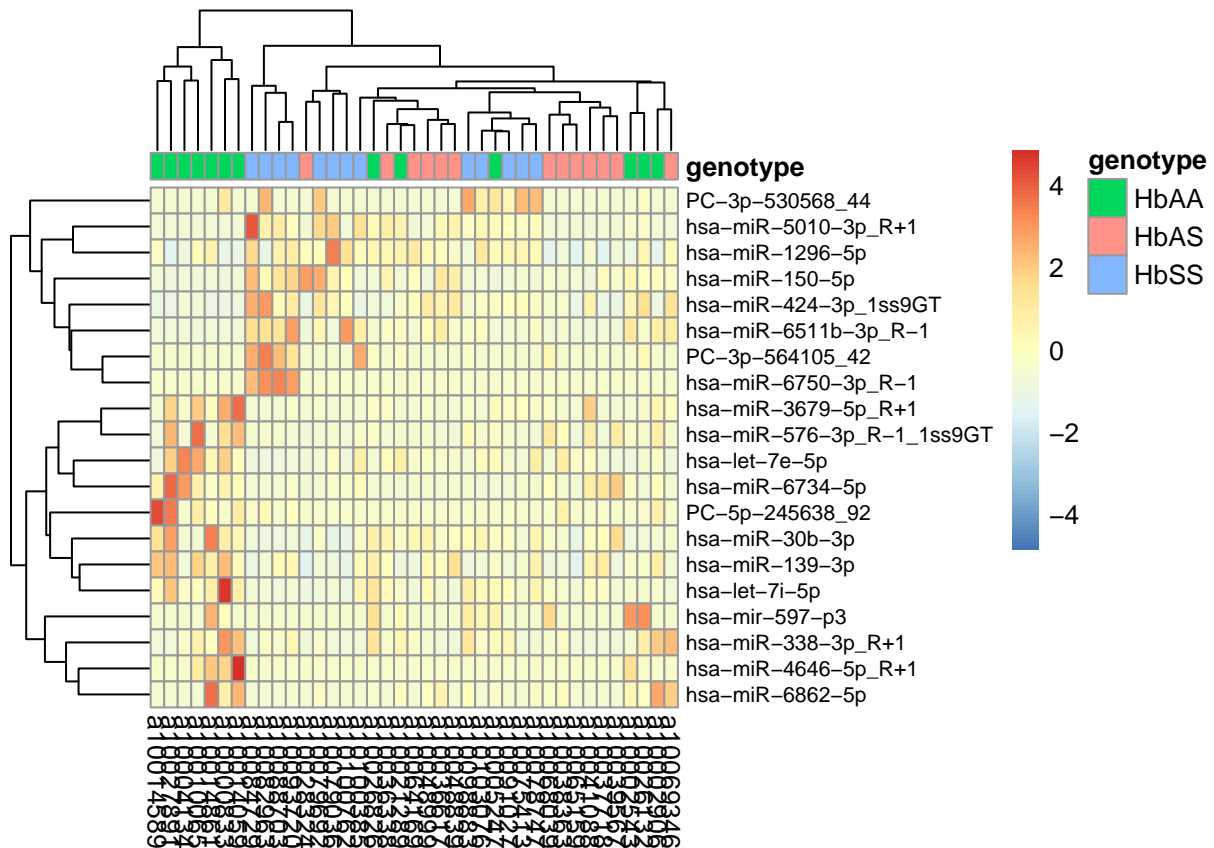
```
results_SS_tbl <- results_SS %>%
  rownames_to_column(var = "miRNA") %>%
  as_tibble()

top_miRNAs <- results_SS_tbl %>%
  arrange(adj.P.Val) %>%
  dplyr::slice(1:20) %>%
  pull(miRNA)

heatmap_matrix <- expr_matrix[top_miRNAs, ]
sample_info_annot <- sample_info %>% column_to_rownames("sample_id")

pheatmap(heatmap_matrix,
  scale = "row",
  annotation_col = sample_info_annot,
  clustering_distance_cols = "euclidean",
```

```
clustering_method = "complete",
show_rownames = TRUE,
fontsize_row = 8)
```



miRNA Target Prediction Using multiMiR

```
if (!"multiMiR" %in% installed) BiocManager::install("multiMiR")
library(multiMiR)
```

```
## Welcome to multiMiR.
##
## multiMiR database URL has been set to the
## default value: http://multimir.org/
##
## Database Version: 2.4.0 Updated: 2024-08-28
```

```
validated_targets <- get_multimir(mirna = top_miRNAs,
                                   table = "validated",
                                   summary = TRUE)
```

```
## Searching mirecords ...
## Searching mirtarbase ...
## Searching tarbase ...
```



```
validated_df <- as.data.frame(validated_targets@data)
```

Extract Unique Gene Symbols

```
target_genes <- validated_df %>%  
  pull(target_symbol) %>%  
  unique()
```

GO Enrichment Analysis

```
if (!"clusterProfiler" %in% installed) BiocManager::install("clusterProfiler")  
if (!"org.Hs.eg.db" %in% installed) BiocManager::install("org.Hs.eg.db")  
  
library(clusterProfiler)  
  
##  
  
## clusterProfiler v4.14.6 Learn more at https://yulab-smu.top/contribution-knowledge-mining/  
##  
## Please cite:  
##  
## G Yu. Thirteen years of clusterProfiler. The Innovation. 2024,  
## 5(6):100722  
  
##  
## Attaching package: 'clusterProfiler'  
  
## The following object is masked from 'package:multiMiR':  
##  
##      select  
  
## The following object is masked from 'package:stats':  
##  
##      filter  
  
library(org.Hs.eg.db)  
  
## Loading required package: AnnotationDbi  
  
## Loading required package: stats4  
  
## Loading required package: BiocGenerics  
  
##  
## Attaching package: 'BiocGenerics'
```

```

## The following object is masked from 'package:limma':
##
##   plotMA

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

## 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, intersect, is.unsorted, lapply, Map, mapply,
##   match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##   Position, rank, rbind, Reduce, rownames, sapply, saveRDS, setdiff,
##   table, tapply, union, unique, unsplit, which.max, which.min

## 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)"'.

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:clusterProfiler':
##
##   rename

## The following object is masked from 'package:tidyr':
##
##   expand

## The following objects are masked from 'package:dplyr':
##
##   first, rename

## The following object is masked from 'package:utils':
##
##   findMatches

```

```

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:clusterProfiler':
##
##     slice

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

##
## Attaching package: 'AnnotationDbi'

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

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

##

gene_df <- bitr(target_genes,
               fromType = "SYMBOL",
               toType = "ENTREZID",
               OrgDb = org.Hs.eg.db)

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

## Warning in bitr(target_genes, fromType = "SYMBOL", toType = "ENTREZID", : 0.2%
## of input gene IDs are fail to map...

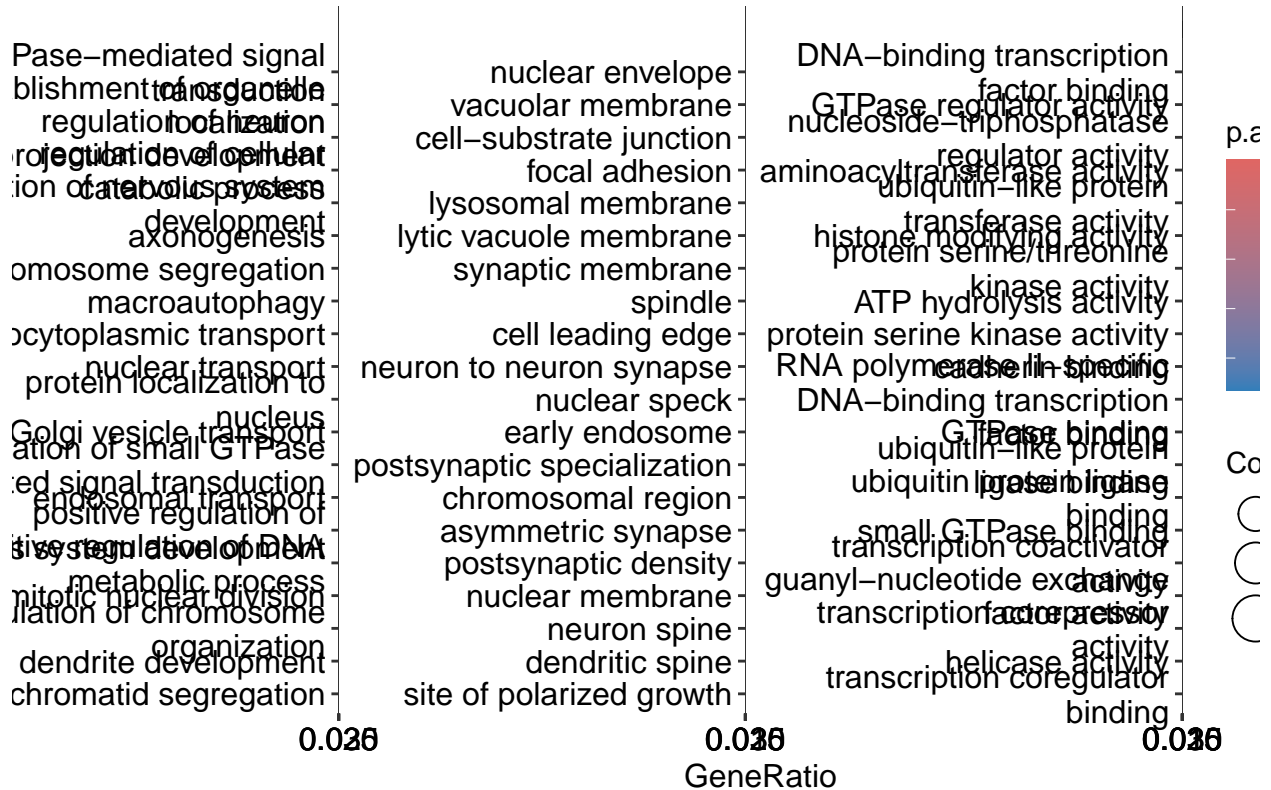
entrez_genes <- unique(gene_df$ENTREZID)

go_results <- enrichGO(gene = entrez_genes,
                      OrgDb = org.Hs.eg.db,
                      keyType = "ENTREZID",
                      ont = "ALL",
                      pAdjustMethod = "BH",
                      pvalueCutoff = 0.05,
                      qvalueCutoff = 0.2,
                      readable = TRUE)

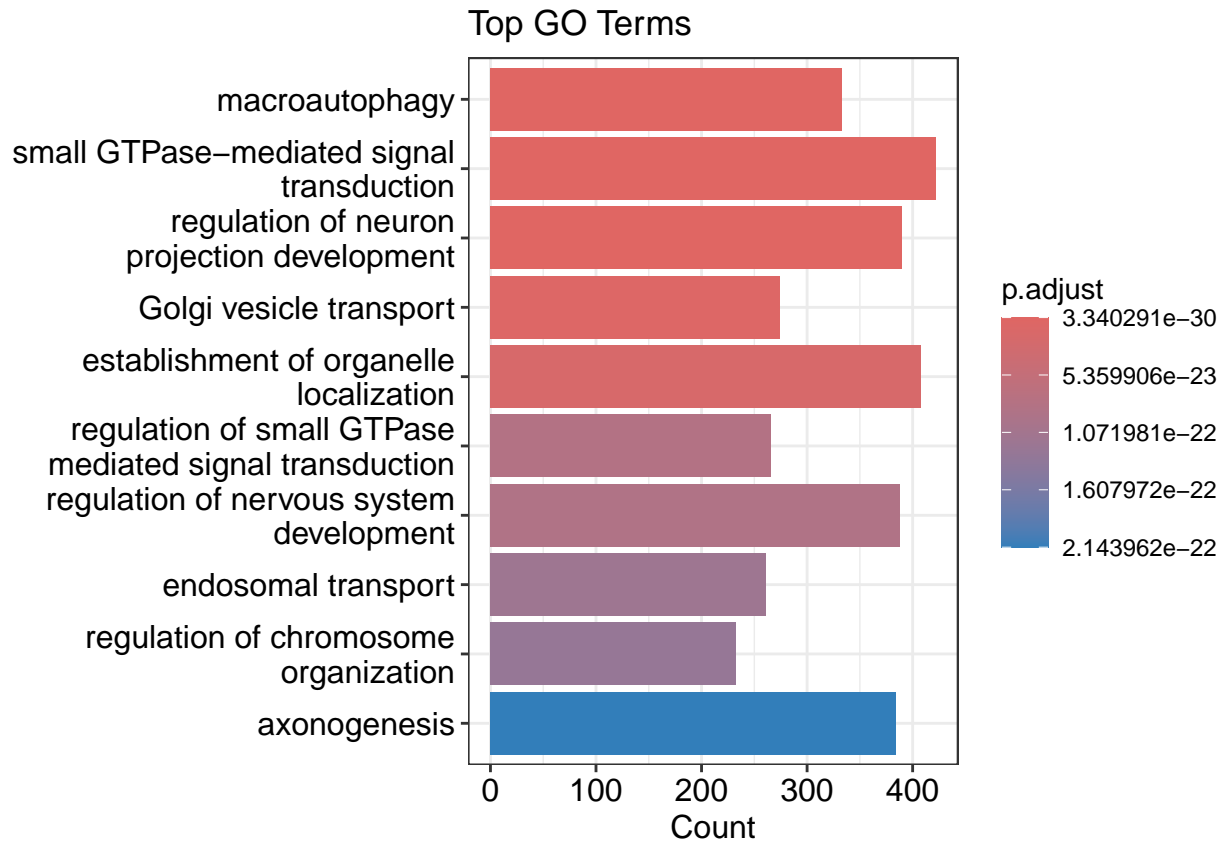
```

Visualize GO Enrichment

```
dotplot(go_results, showCategory = 20, split = "ONTOLOGY") +
  ggplot2::facet_wrap(~ONTOLOGY, scales = "free")
```

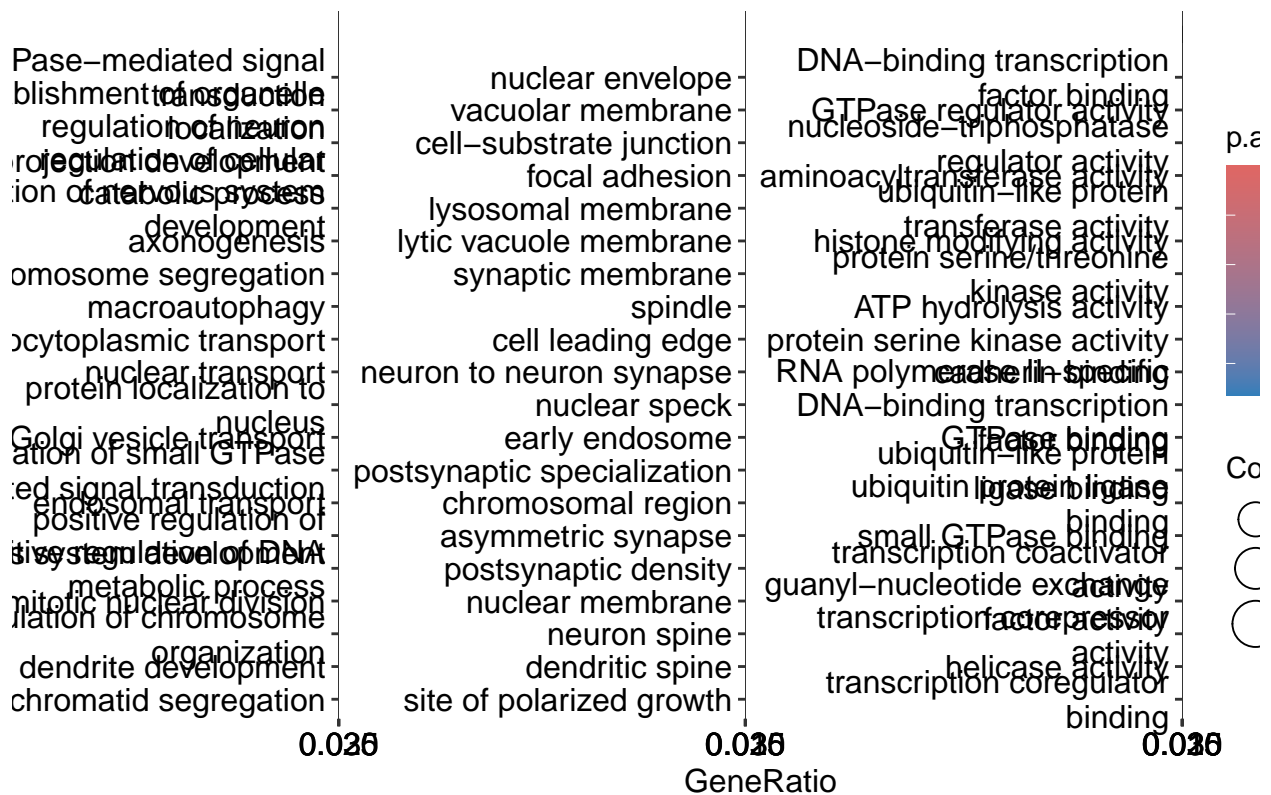


```
barplot(go_results, showCategory = 10, title = "Top GO Terms")
```

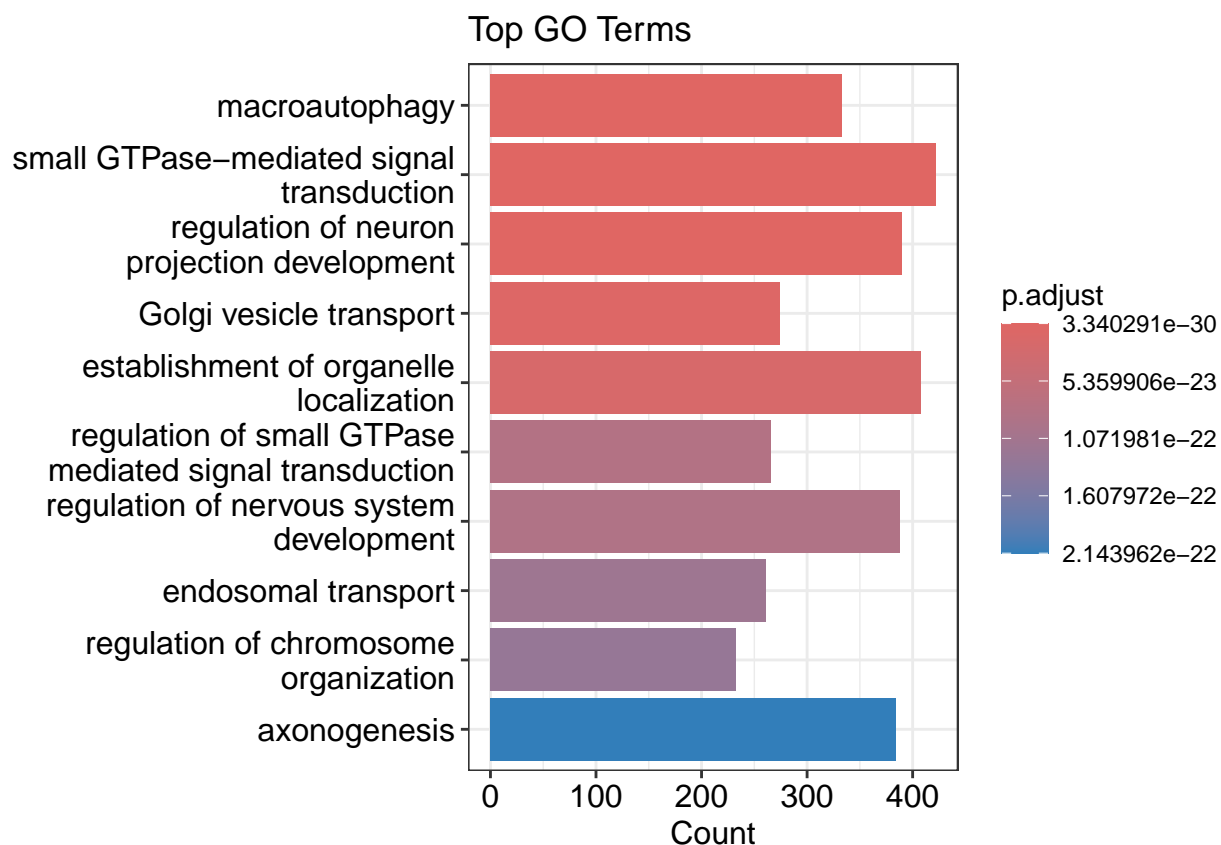


Inflammation-Related GO Terms

```
# Dotplot and barplot
dotplot(go_results, showCategory = 20, split = "ONTOLOGY") +
  ggplot2::facet_wrap(~ONTOLOGY, scales = "free")
```



```
barplot(go_results, showCategory = 10, title = "Top GO Terms")
```



```
# Filter inflammatory/immune-related terms
inflammatory_terms <- go_results@result %>%
  filter(grepl("inflamm|immune|cytokine|macrophage", Description, ignore.case = TRUE)) %>%
  dplyr::select(ID, Description, Count, p.adjust)

inflammatory_terms
```

```
##          ID
## GO:0061640 GO:0061640
## GO:0000910 GO:0000910
## GO:0000281 GO:0000281
## GO:0032506 GO:0032506
## GO:0002218 GO:0002218
## GO:0002758 GO:0002758
## GO:0002520 GO:0002520
## GO:0002200 GO:0002200
## GO:1902410 GO:1902410
## GO:0045089 GO:0045089
## GO:0045088 GO:0045088
## GO:0002562 GO:0002562
## GO:0061082 GO:0061082
## GO:0032465 GO:0032465
## GO:0061081 GO:0061081
## GO:1903490 GO:1903490
## GO:0002440 GO:0002440
```

G0:0002433 G0:0002433
 ## G0:0002204 G0:0002204
 ## G0:0002208 G0:0002208
 ## G0:0002720 G0:0002720
 ## G0:0002381 G0:0002381
 ## G0:0140374 G0:0140374
 ## G0:0002702 G0:0002702
 ## G0:0010934 G0:0010934
 ## G0:0010935 G0:0010935
 ## G0:2000317 G0:2000317
 ## G0:0002718 G0:0002718
 ## G0:0072538 G0:0072538
 ## G0:0141087 G0:0141087
 ## G0:0002367 G0:0002367
 ## G0:0002287 G0:0002287
 ## G0:0002293 G0:0002293
 ## G0:0060907 G0:0060907
 ## G0:0060760 G0:0060760
 ## G0:0060759 G0:0060759
 ## G0:0002700 G0:0002700
 ## G0:1900227 G0:1900227
 ## G0:0002294 G0:0002294

	Descript.
## G0:0061640	cytoskeleton-dependent cytokines
## G0:0000910	cytokines
## G0:0000281	mitotic cytokines
## G0:0032506	cytokinetic process
## G0:0002218	activation of innate immune response
## G0:0002758	innate immune response-activating signaling pathway
## G0:0002520	immune system development
## G0:0002200	somatic diversification of immune receptors
## G0:1902410	mitotic cytokinetic process
## G0:0045089	positive regulation of innate immune response
## G0:0045088	regulation of innate immune response
## G0:0002562	somatic diversification of immune receptors via germline recombination within a single locus
## G0:0061082	myeloid leukocyte cytokine production
## G0:0032465	regulation of cytokines
## G0:0061081	positive regulation of myeloid leukocyte cytokine production involved in immune response
## G0:1903490	positive regulation of mitotic cytokines
## G0:0002440	production of molecular mediator of immune response
## G0:0002433	immune response-regulating cell surface receptor signaling pathway involved in phagocytosis
## G0:0002204	somatic recombination of immunoglobulin genes involved in immune response
## G0:0002208	somatic diversification of immunoglobulins involved in immune response
## G0:0002720	positive regulation of cytokine production involved in immune response
## G0:0002381	immunoglobulin production involved in immunoglobulin-mediated immune response
## G0:0140374	antiviral innate immune response
## G0:0002702	positive regulation of production of molecular mediator of immune response
## G0:0010934	macrophage cytokine production
## G0:0010935	regulation of macrophage cytokine production
## G0:2000317	negative regulation of T-helper 17 type immune response
## G0:0002718	regulation of cytokine production involved in immune response
## G0:0072538	T-helper 17 type immune response
## G0:0141087	positive regulation of inflammasome-mediated signaling pathway
## G0:0002367	cytokine production involved in immune response

## G0:0002287		alpha-beta T cell activation involved in immune respon
## G0:0002293		alpha-beta T cell differentiation involved in immune respon
## G0:0060907		positive regulation of macrophage cytokine product
## G0:0060760		positive regulation of response to cytokine stimu
## G0:0060759		regulation of response to cytokine stimu
## G0:0002700		regulation of production of molecular mediator of immune respon
## G0:1900227		positive regulation of NLRP3 inflammasome complex assem
## G0:0002294		CD4-positive, alpha-beta T cell differentiation involved in immune respon
##	Count	p.adjust
## G0:0061640	112	7.928272e-12
## G0:0000910	163	5.965915e-11
## G0:0000281	86	1.515904e-09
## G0:0032506	45	1.886736e-05
## G0:0002218	231	1.379053e-04
## G0:0002758	217	1.781532e-04
## G0:0002520	151	3.954029e-04
## G0:0002200	64	5.428203e-04
## G0:1902410	31	7.482604e-04
## G0:0045089	273	9.707481e-04
## G0:0045088	326	1.681205e-03
## G0:0002562	58	2.408391e-03
## G0:0061082	42	2.677003e-03
## G0:0032465	74	2.835447e-03
## G0:0061081	30	7.471662e-03
## G0:1903490	13	7.528573e-03
## G0:0002440	163	9.031435e-03
## G0:0002433	20	9.700378e-03
## G0:0002204	44	1.113436e-02
## G0:0002208	44	1.113436e-02
## G0:0002720	64	1.906083e-02
## G0:0002381	46	1.976268e-02
## G0:0140374	46	1.976268e-02
## G0:0002702	101	2.061701e-02
## G0:0010934	31	2.295888e-02
## G0:0010935	31	2.295888e-02
## G0:2000317	14	2.312593e-02
## G0:0002718	91	2.648806e-02
## G0:0072538	42	2.748642e-02
## G0:0141087	19	3.279145e-02
## G0:0002367	91	3.494361e-02
## G0:0002287	62	3.791824e-02
## G0:0002293	62	3.791824e-02
## G0:0060907	21	4.036585e-02
## G0:0060760	51	4.081604e-02
## G0:0060759	138	4.206049e-02
## G0:0002700	136	4.262139e-02
## G0:1900227	18	4.419597e-02
## G0:0002294	61	4.482191e-02

Network Visualization of miRNA–Gene–GO

```
library(igraph)
```

```
##  
## Attaching package: 'igraph'  
  
## The following object is masked from 'package:IRanges':  
##  
##      union  
  
## The following object is masked from 'package:S4Vectors':  
##  
##      union  
  
## The following objects are masked from 'package:BiocGenerics':  
##  
##      normalize, path, union  
  
## The following object is masked from 'package:clusterProfiler':  
##  
##      simplify  
  
## The following object is masked from 'package:tibble':  
##  
##      as_data_frame  
  
## The following object is masked from 'package:tidyr':  
##  
##      crossing  
  
## The following objects are masked from 'package:dplyr':  
##  
##      as_data_frame, groups, union  
  
## The following objects are masked from 'package:stats':  
##  
##      decompose, spectrum  
  
## The following object is masked from 'package:base':  
##  
##      union
```

```
library(ggraph)
```

```
# Get genes in inflammatory GO terms  
inflamm_gene_ids <- unlist(go_results@geneSets[inflammatory_terms$ID])  
inflamm_genes <- bitr(inflamm_gene_ids, fromType="ENTREZID", toType="SYMBOL", OrgDb=org.Hs.eg.db)$SYMBOL
```

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

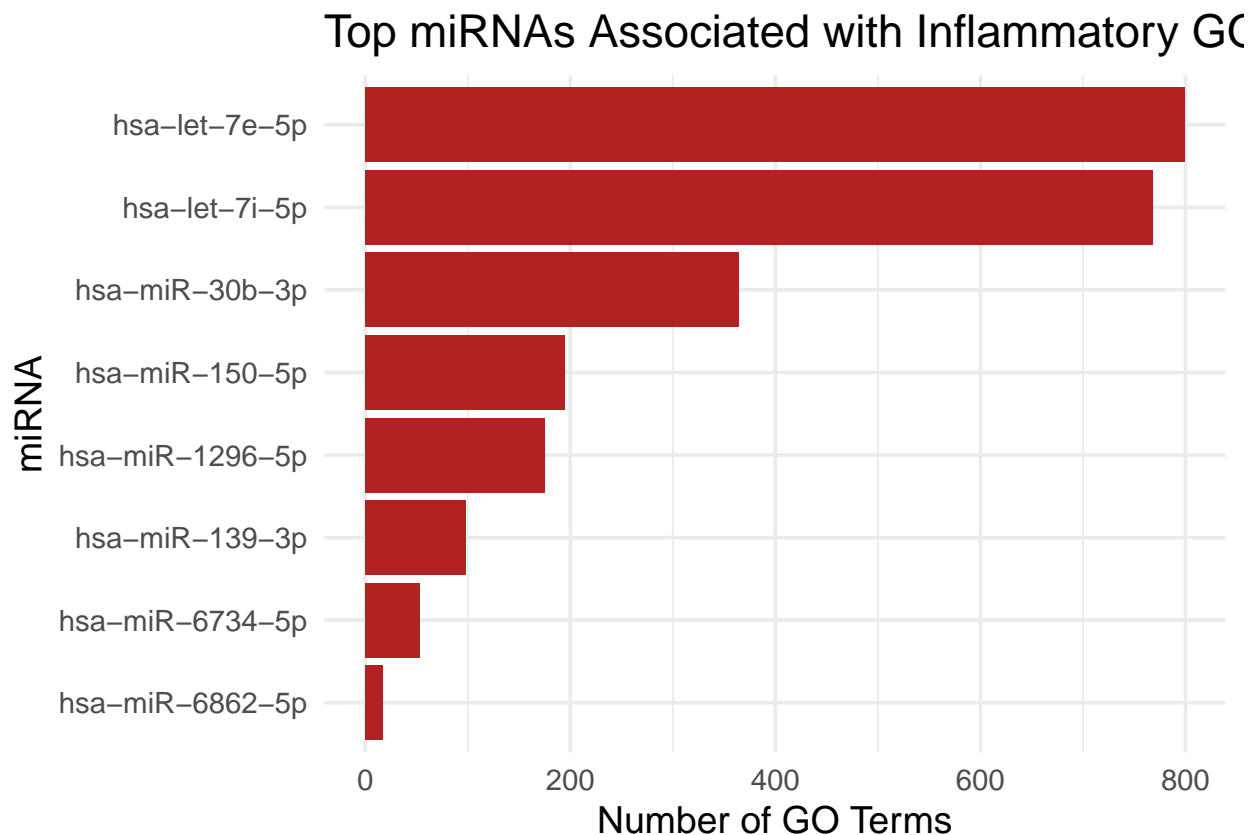
```

# Subset validated targets to only those
inflamm_targets <- validated_df %>%
  filter(target_symbol %in% inflamm_genes)

# Count how many inflammatory GO terms each miRNA is involved in
mirna_go_counts <- inflamm_targets %>%
  count(mature_mirna_id, name = "Num_Inflammatory_GO_Terms") %>%
  arrange(desc(Num_Inflammatory_GO_Terms))

# Plot top miRNAs associated with inflammation
ggplot(mirna_go_counts, aes(x = reorder(mature_mirna_id, Num_Inflammatory_GO_Terms),
  y = Num_Inflammatory_GO_Terms)) +
  geom_col(fill = "firebrick") +
  coord_flip() +
  theme_minimal(base_size = 14) +
  labs(title = "Top miRNAs Associated with Inflammatory GO Terms",
    x = "miRNA",
    y = "Number of GO Terms")

```



```

library(knitr)
top_mirna_names <- mirna_go_counts %>% dplyr::slice(1:5) %>% pull(mature_mirna_id)

presentation_table <- inflamm_targets %>%
  filter(mature_mirna_id %in% top_mirna_names) %>%
  distinct(mature_mirna_id, target_symbol) %>%

```

```

dplyr::left_join(gene_df, by = c("target_symbol" = "SYMBOL")) %>%
dplyr::left_join(go_results@result, by = c("ENTREZID" = "geneID")) %>%
dplyr::select(mature_mirna_id, target_symbol) %>%
arrange(mature_mirna_id)

kable(head(presentation_table, 10),
caption = "Sample of Inflammatory GO Terms Linked to Top miRNAs")

```

Table 1: Sample of Inflammatory GO Terms Linked to Top miRNAs

mature_mirna_id	target_symbol
hsa-let-7e-5p	DHX15
hsa-let-7e-5p	PTK2
hsa-let-7e-5p	IL6R
hsa-let-7e-5p	MATR3
hsa-let-7e-5p	HMGB1
hsa-let-7e-5p	POLR3D
hsa-let-7e-5p	AURKB
hsa-let-7e-5p	YWHAZ
hsa-let-7e-5p	PLK1
hsa-let-7e-5p	STAT3

```

# Subset validated targets for top 5 miRNAs
top_validated_targets <- validated_df %>%
  filter(mature_mirna_id %in% top_mirna_names) %>%
  dplyr::select(mature_mirna_id, target_symbol) %>%
  distinct()

# Count how many of the top miRNAs target each gene
top_genes_by_target_count <- top_validated_targets %>%
  group_by(target_symbol) %>%
  summarise(num_top_miRNAs_targeting = n()) %>%
  arrange(desc(num_top_miRNAs_targeting))

# Show top 10 most commonly targeted genes
knitr::kable(head(top_genes_by_target_count, 100),
caption = "Genes Most Frequently Targeted by Top Differentially Expressed miRNAs")

```

Table 2: Genes Most Frequently Targeted by Top Differentially Expressed miRNAs

target_symbol	num_top_miRNAs_targeting
ACLY	5
ACTR2	5
ADGRL1	5
ADNP	5
AGO2	5
AKIRIN1	5

target_symbol	num_top_miRNAs_targeting
ANKFY1	5
AP2B1	5
ASH1L	5
ATP2A2	5
ATXN7L3	5
BAZ2A	5
BCAS3	5
BCL9	5
BRAF	5
BUB3	5
CAND1	5
CANX	5
CAPRIN1	5
CBX6	5
CD2BP2	5
CDK2	5
CEP126	5
CIC	5
CPSF6	5
CS	5
DCAF8	5
DDB1	5
DDX6	5
DHTKD1	5
DNAJC6	5
DYNC1H1	5
EI24	5
EIF3A	5
EIF4A1	5
EIF4G1	5
EIF4G2	5
EPHB2	5
ETF1	5
F2RL1	5
FASN	5
FBXL20	5
FBXO41	5
FBXW2	5
FER	5
FLNA	5
FNIP1	5
FOKK1	5
FOKK2	5
FRYL	5
GATAD2B	5
GFM1	5
GIGYF2	5
GNB1	5
HDGF	5
HDLBP	5
HECTD4	5
HERC1	5

target_symbol	num_top_miRNAs_targeting
HK1	5
HNRNPA2B1	5
HNRNPF	5
HNRNPL	5
HSPA4	5
HUWE1	5
IGSF8	5
INTS7	5
ITPK1	5
KHSRP	5
KIAA1549	5
KIAA2013	5
KIF5B	5
KLHL21	5
KPNA6	5
LDHB	5
LRP1	5
LY6G5B	5
MACF1	5
MAP4	5
MARCKSL1	5
MAST3	5
MCL1	5
MED13	5
MED13L	5
MED14	5
METAP2	5
MLEC	5
MON1B	5
MRPL17	5
MYH9	5
NCS1	5
NFAT5	5
NOTCH2	5
NUFIP2	5
PARD3	5
PFKFB3	5
PHF12	5
PICALM	5
PPFIA1	5
PPP2R1B	5

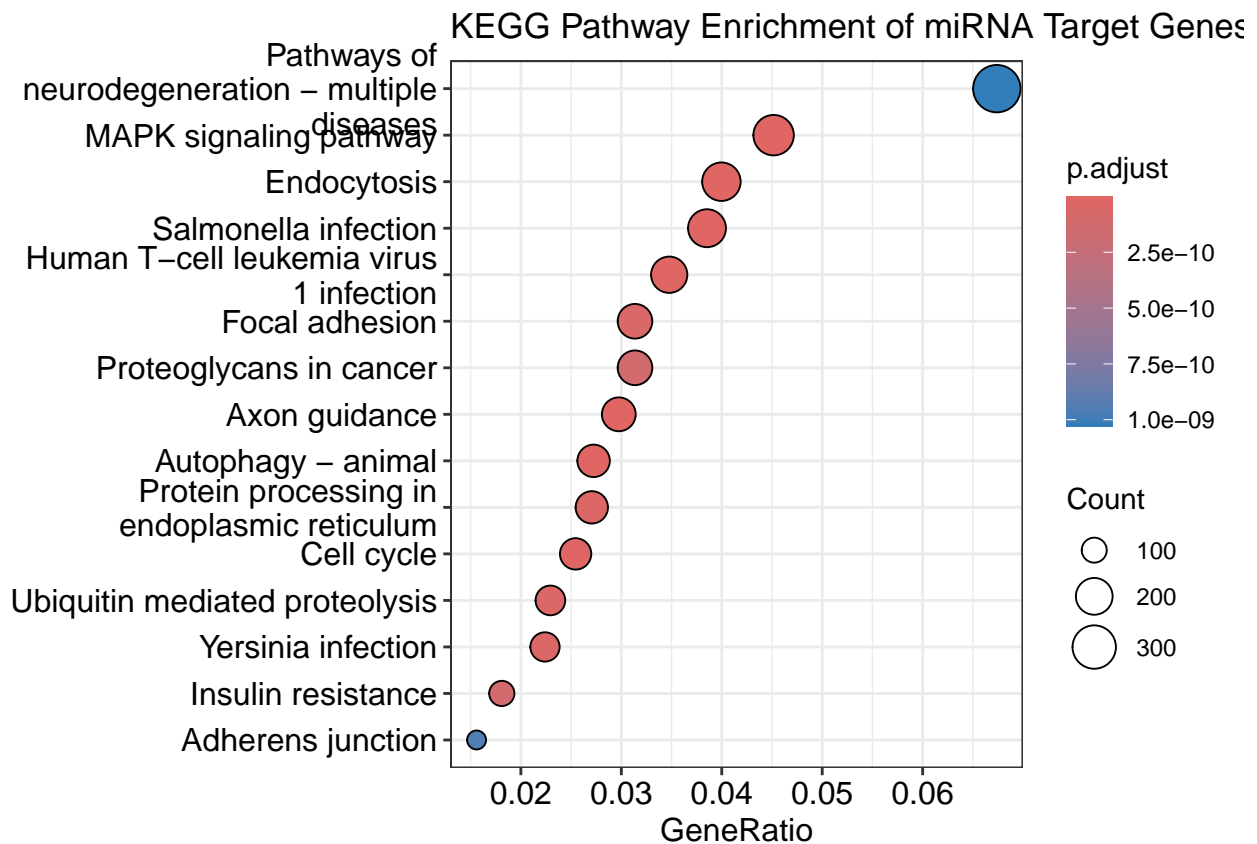
```
# KEGG Pathway Enrichment
```

```
kegg_results <- enrichKEGG(gene = entrez_genes,
  organism = "hsa",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.05,
  qvalueCutoff = 0.2)
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/link/hsa/pathway"...
```

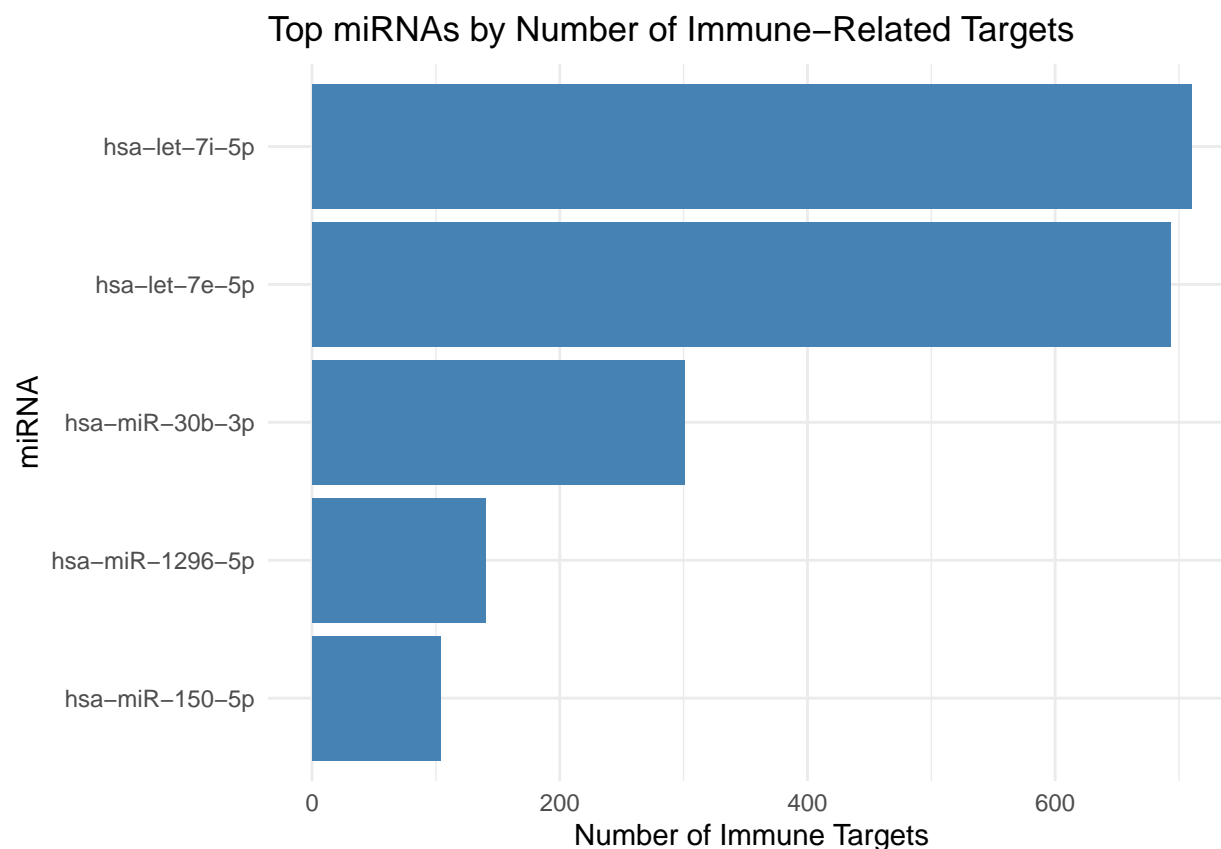
```
## Reading KEGG annotation online: "https://rest.kegg.jp/list/pathway/hsa"...
```

```
# Visualize top pathways
dotplot(kegg_results, showCategory = 15) +
  ggtitle("KEGG Pathway Enrichment of miRNA Target Genes")
```



```
# Summarize counts per miRNA per immune pathway
immune_target_matrix <- inflamm_targets %>%
  filter(mature_mirna_id %in% top_mirna_names) %>%
  group_by(mature_mirna_id) %>%
  summarise(Num_Immune_Genes = n_distinct(target_symbol))

ggplot(immune_target_matrix, aes(x = reorder(mature_mirna_id, Num_Immune_Genes),
  y = Num_Immune_Genes)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Top miRNAs by Number of Immune-Related Targets",
    x = "miRNA", y = "Number of Immune Targets") +
  theme_minimal()
```



```
# Identify shared genes across multiple miRNAs
shared_genes <- top_validated_targets %>%
  group_by(target_symbol) %>%
  summarise(miRNA_Count = n()) %>%
  filter(miRNA_Count > 1) %>%
  arrange(desc(miRNA_Count))

knitr::kable(head(shared_genes, 10),
  caption = "Shared Target Genes Across Multiple Top miRNAs")
```

Table 3: Shared Target Genes Across Multiple Top miRNAs

target_symbol	miRNA_Count
	5
ACLY	5
ACTR2	5
ADGRL1	5
ADNP	5
AGO2	5
AKIRIN1	5
ANKFY1	5
AP2B1	5
ASH1L	5