GSE Analysis

Hussein

2025-07-08

Load the Necessary Libraries

```r
# Install and load required packages
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("GO.db")
BiocManager::install("org.Hs.eg.db")
BiocManager::install("biomaRt")
BiocManager::install("ReactomePA")
BiocManager::install("enrichplot")
BiocManager::install("clusterProfiler")

library(GO.db)
library(org.Hs.eg.db)
library(biomaRt)
library(enrichplot)
library(ReactomePA)
library(clusterProfiler)
library(tidyverse)
```

Load the Data Sets

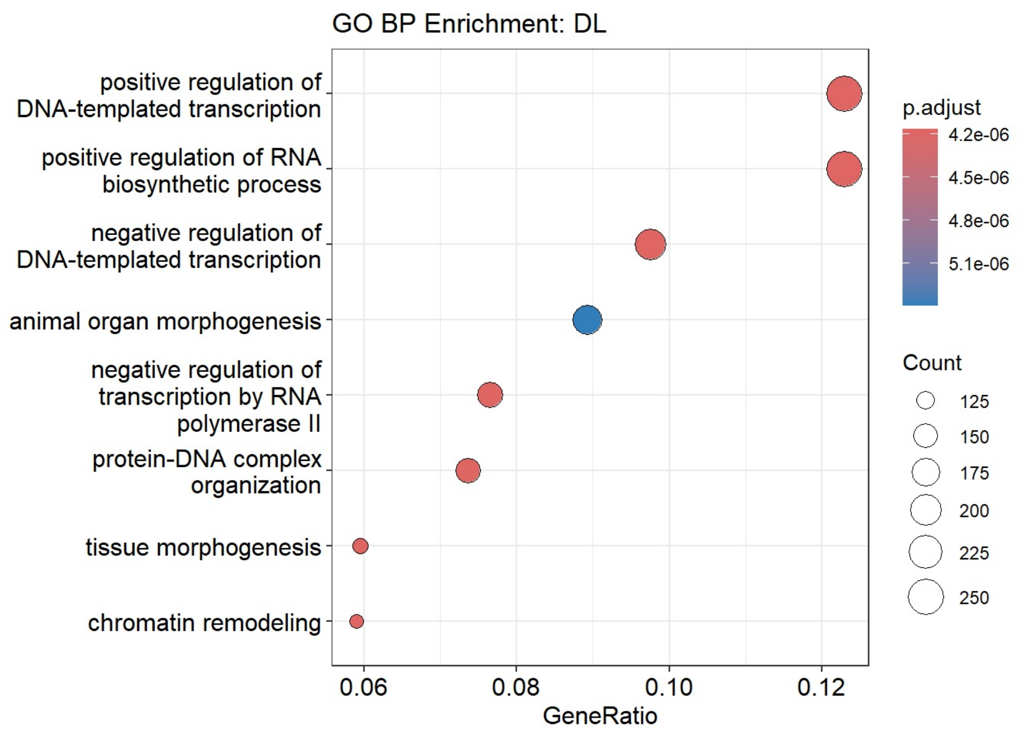
```r
# Load list of human protein-coding genes
protein_coding_genes <- read_delim("C:/Users/HP-ssd/Desktop/Short term project/protein coding genes/gene_with_pro
tein_product.txt",
                                   delim = "    ", escape_double = FALSE,
                                   trim_ws = TRUE)

protein_coding_genes_list <- protein_coding_genes$symbol

# Load FUSIL data
fusil_m_gene <-  read_delim("C:/Users/HP-ssd/Desktop/Short term project2/fusil.csv")

# Load human gene paralogues from BioMart
human_gene_paralogues <- read.csv("C:/Users/HP-ssd/Desktop/Short term project2/paralogues/human_gene_paralogues.c
sv")

# Clean and rename columns
human_gene_paralogues <- human_gene_paralogues %>%
  select(-1, -2, -4) %>%
  rename(gene_symbol = external_gene_name)
```

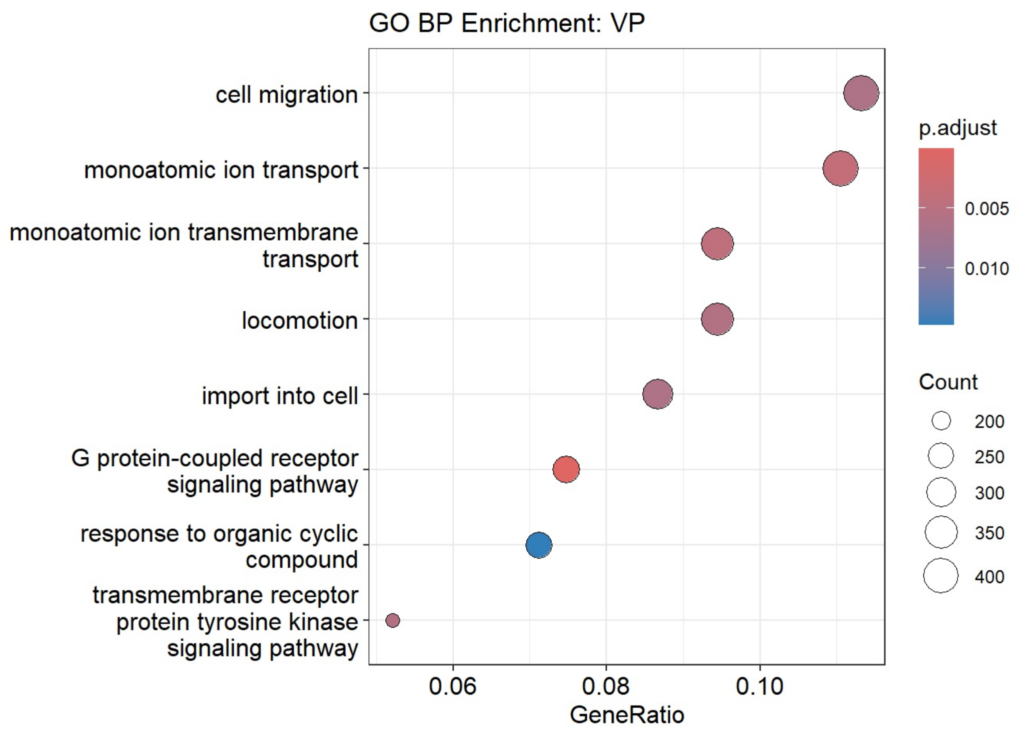
Create a Matrix of FUSIL Categories for Genes and Paralogues

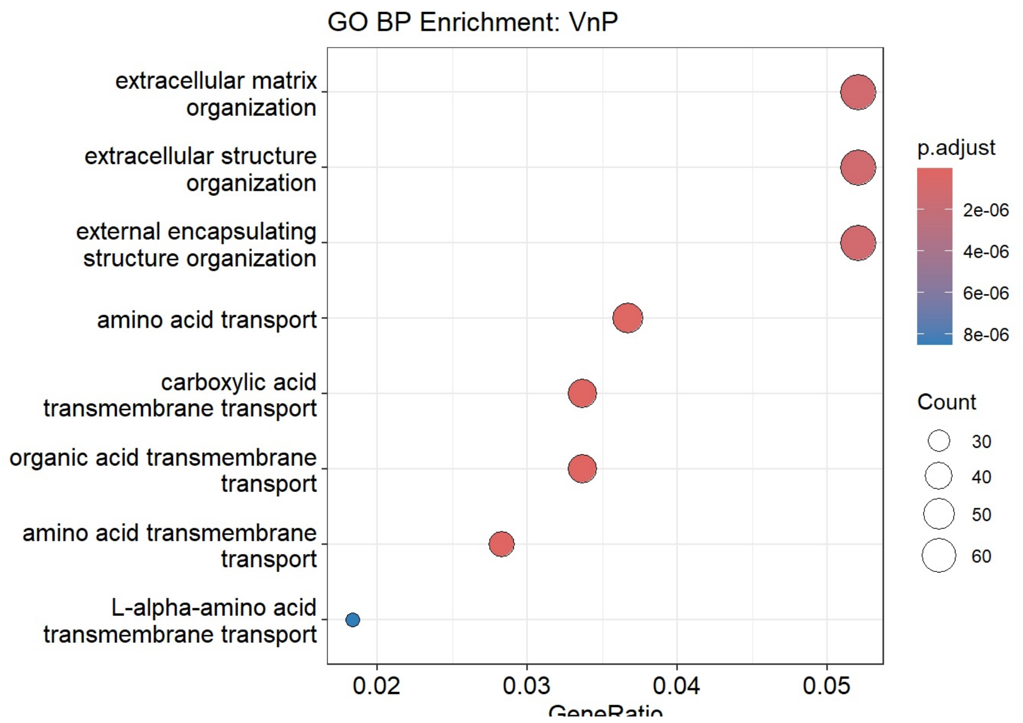
```r
# Keep necessary columns from fusil data
Fusil_genes <- fusil_m_gene %>%
  select(-1, -2)

# Join paralogues with fusil annotations
Fusil_genes_paralogues <- human_gene_paralogues %>%
  left_join(Fusil_genes, by = c("hsapiens_paralog_associated_gene_name" = "gene_symbol")) %>%
  filter(hsapiens_paralog_associated_gene_name %in% protein_coding_genes_list) %>%
  rename("fusil_paralogue" = "fusil")

# Join with fusil for the main gene
Fusil_genes_paralogues <- Fusil_genes_paralogues %>%
  left_join(Fusil_genes, by = c("gene_symbol" = "gene_symbol"))

# Clean and rearrange
Fusil_genes_paralogues <- Fusil_genes_paralogues %>%
  relocate(fusil, .after = "gene_symbol") %>%
  relocate(fusil_paralogue, .after = "hsapiens_paralog_associated_gene_name") %>%
  na.omit()

# Annotate FUSIL match and similarity bin
fusil_match <- Fusil_genes_paralogues %>%
  mutate(FUSIL_match = ifelse(fusil == fusil_paralogue, "Match", "Mismatch")) %>%
  mutate(SIMILARITY_bin = case_when(
    hsapiens_paralog_perc_id >= 80 ~ "High >80% ",
    hsapiens_paralog_perc_id >= 60 ~ "Medium-High 60-80%",
    hsapiens_paralog_perc_id >= 40 ~ "Medium 40-60%",
    hsapiens_paralog_perc_id >= 20 ~ "Medium-Low 20-50%",
    TRUE ~ "Low <20%"))
```

Perform GSE Analysis Using FUSIL Genes as Universe

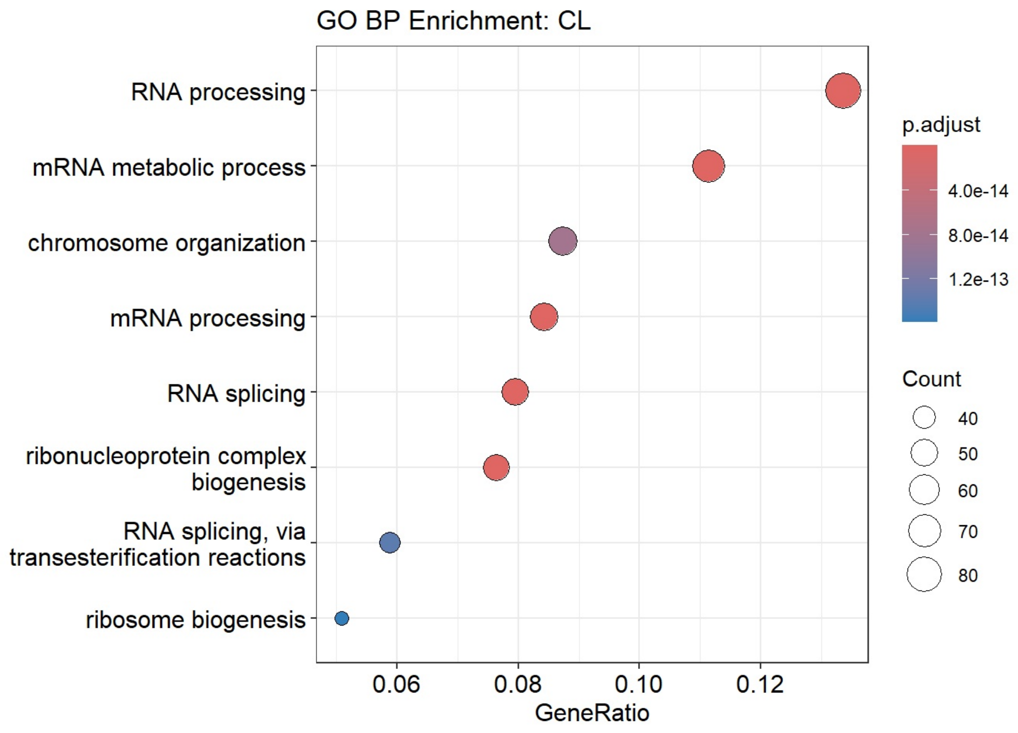
```r
# Set up BioMart
hs_mart <- useMart(dataset = "hsapiens_gene_ensembl", biomart = "ensembl")

# Get reference gene set (universe)
genes <- unique(fusil_match$gene_symbol)

gene_entrez_id <- getBM(
  attributes = c('hgnc_symbol', 'ensembl_gene_id', 'entrezgene_id'),
  filters = 'hgnc_symbol',
  values = genes,
  mart = hs_mart
)

reference_set_entrez <- unique(gene_entrez_id$entrezgene_id)
reference_set_entrez <- reference_set_entrez[!is.na(reference_set_entrez)]
reference_set_entrez <- as.character(reference_set_entrez)

# Prepare enrichment results list
categories <- unique(fusil_match$fusil)
enrichment_results_list <- list()

# Loop through each FUSIL bin
for (cat in categories) {
  message(paste(" Processing category:", cat))

  matching_subset <- fusil_match %>%
    filter(fusil == cat)

  gene_match <- unique(matching_subset$hsapiens_paralog_associated_gene_name)

  gene_mapping <- getBM(
    attributes = c('hgnc_symbol', 'ensembl_gene_id', 'entrezgene_id'),
    filters = 'hgnc_symbol',
    values = gene_match,
    mart = hs_mart
  )

  gene_set_entrez <- unique(gene_mapping$entrezgene_id)
  gene_set_entrez <- gene_set_entrez[!is.na(gene_set_entrez)]
  gene_set_entrez <- gene_set_entrez[gene_set_entrez %in% reference_set_entrez]

  message(paste(" Genes in test set:", length(gene_set_entrez)))

  if (length(gene_set_entrez) < 5) {
    message(paste("⚠  Too few genes for enrichment in category:", cat, "- skipping."))
    next
  }

  enrichment_result <- enrichGO(
    gene = gene_set_entrez,
    universe = reference_set_entrez,
    OrgDb = org.Hs.eg.db,
    ont = "BP",
    pAdjustMethod = "BH",
    readable = TRUE
  )

  if (is.null(enrichment_result) || nrow(as.data.frame(enrichment_result)) == 0) {
    message(paste("⚠  No enrichment found for category:", cat, "- skipping plot."))
    next
  }

  enrichment_results_list[[cat]] <- enrichment_result

  p <- dotplot(enrichment_result, showCategory = 8) +
    ggtitle(paste("GO BP Enrichment:", cat))

  message(paste("✅ Analysis done for category:", cat))
  print(p)
}
```

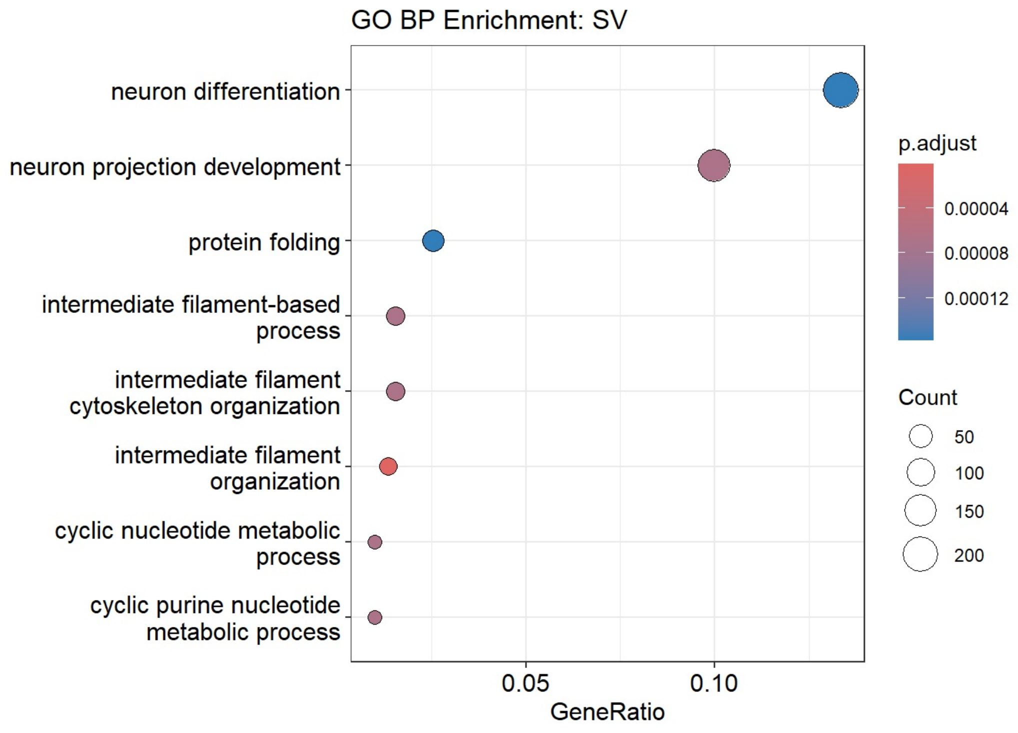

GO BP Enrichment: DL
positive regulation of DNA-templated transcription
positive regulation of RNA biosynthetic process
negative regulation of DNA-templated transcription
animal organ morphogenesis
negative regulation of transcription by RNA polymerase II
protein-DNA complex organization
tissue morphogenesis
chromatin remodeling
0.06
0.08
0.10
0.12
GeneRatio
p.adjust
4.2e-06
4.5e-06
4.8e-06
5.1e-06
Count
125
150
175
200
225
250


GO BP Enrichment: VP
cell migration
monoatomic ion transport
monoatomic ion transmembrane transport
locomotion
import into cell
G protein-coupled receptor signaling pathway
response to organic cyclic compound
transmembrane receptor protein tyrosine kinase signaling pathway
0.06
0.08
0.10
GeneRatio
p.adjust
0.005
0.010
Count
200
250
300
350
400


GO BP Enrichment: VnP
extracellular matrix organization
extracellular structure organization
external encapsulating structure organization
amino acid transport
carboxylic acid transmembrane transport
organic acid transmembrane transport
amino acid transmembrane transport
L-alpha-amino acid transmembrane transport
0.02
0.03
0.04
0.05
GeneRatio
p.adjust
2e-06
4e-06
6e-06
8e-06
Count
30
40
50
60

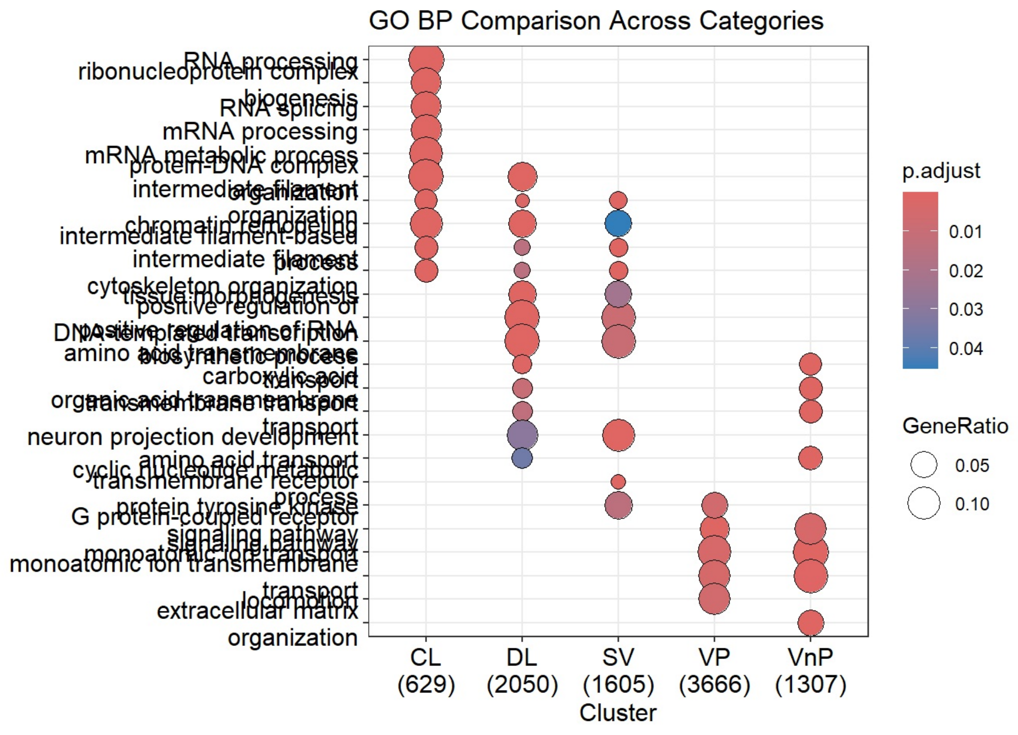
GeneRatio





Compare All Categories in One Plot

```r
# Group paralogues by FUSIL bin
paralogue_lists <- fusil_match %>%
  group_by(fusil) %>%
  summarise(paralogues = list(unique(hsapiens_paralog_associated_gene_name))) %>%
  deframe()

# Map each group to Entrez IDs
entrez_sets <- lapply(paralogue_lists, function(gene_symbols) {
  gene_mapping <- getBM(
    attributes = c('hgnc_symbol', 'entrezgene_id'),
    filters = 'hgnc_symbol',
    values = gene_symbols,
    mart = hs_mart
  )
  entrez_ids <- unique(gene_mapping$entrezgene_id)
  entrez_ids <- entrez_ids[!is.na(entrez_ids)]
  entrez_ids[entrez_ids %in% reference_set_entrez]
})

# Remove small gene sets
entrez_sets <- entrez_sets[sapply(entrez_sets, length) >= 5]

# Perform comparative enrichment
compare_result <- compareCluster(
  geneCluster = entrez_sets,
  fun = "enrichGO",
  OrgDb = org.Hs.eg.db,
  ont = "BP",
  universe = reference_set_entrez,
  pAdjustMethod = "BH",
  readable = TRUE
)

# Plot comparison
dotplot(compare_result, showCategory = 5) +
  ggtitle("GO BP Comparison Across Categories")
```


GO BP Comparison Across Categories
RNA processing
ribonucleoprotein complex biogenesis
RNA splicing
mRNA processing
mRNA metabolic process
protein-DNA complex organization
intermediate filament organization
chromatin remodeling
intermediate filament-based process
intermediate filament cytoskeleton organization
tissue morphogenesis
positive regulation of DNA-templated transcription
positive regulation of RNA biosynthetic process
amino acid transmembrane transport
carboxylic acid transmembrane transport
organic acid transmembrane transport
neuron projection development
amino acid transport
cyclic nucleotide metabolic process
transmembrane receptor protein tyrosine kinase signaling pathway
G protein-coupled receptor signaling pathway
monoatomic ion transport
monoatomic ion transmembrane transport
locomotion
extracellular matrix organization
CL (629)
DL (2050)
SV (1605)
VP (3666)
VnP (1307)
Cluster
p.adjust
GeneRatio