

Figure 1f — GSEA IC heatmap (Discovery & Validation)

MAXOMOD_CSF

2026-02-16

Contents

Overview	1
Setup	2
Load packages	2
Project root and parameters	2
Load GSEA results	3
Build GO term \times comparison matrix	3
GO semantic similarity and clustering	4
Heatmap annotation and color scale	4
Draw heatmap and save (no display in report; only final ht used for files)	5
Preview in HTML	6
Export summary CSV	7
Outputs	8

Overview

This document reproduces the **GSEA IC heatmap** step: summarized GSEA results (Discovery & Validation) as a heatmap with GO terms clustered by semantic similarity (Information Content).

Equivalent command line:

```
Rscript 13_GSEA_IC_heatmap.R -i  
↪ Discovery=Discovery/06_GSEA/GSEA_result.rds,Validation=Validation/06_GSEA/GSEA_result.rds  
↪ -o 13_GSEA_IC_heatmap -c 6
```

Inputs are read from `demo/Discovery/06_GSEA/` and `demo/Validation/06_GSEA/` (already copied). Paths are relative to the **project root**. Run the “Project root and parameters” chunk first when running chunks interactively.

Setup

Load packages

```
library(msigdbr)
library(circlize)
library(dplyr)
library(tidyr)
library(ggsci)
library(tidyverse)
library(GOsemSim)
library(dendextend)
library(ComplexHeatmap)
```

Project root and parameters

```
project_root <- if (basename(getwd()) == "vignettes") {
  normalizePath("../", winslash = "/")
} else {
  getwd()
}
setwd(project_root)
message("Project root: ", project_root)
```

```
## Project root: /Users/xliu2942/Documents/Projects/MAXOMOD/MAXOMOD_CSF
```

```
# Parse named input paths and resolve to full paths
parse_named_paths <- function(input_string) {
  parts <- strsplit(input_string, ",")[[1]]
  split_parts <- strsplit(parts, "=")
  names_vec <- sapply(split_parts, `[`, 1)
  paths_vec <- sapply(split_parts, `[`, 2)
  setNames(paths_vec, names_vec)
}

gsea_paths_relative <- parse_named_paths(params$input)
gsea_paths <- setNames(file.path(project_root, gsea_paths_relative),
  ↪ names(gsea_paths_relative))

output_dir <- file.path(project_root, params$output)
k_num <- params$cluster_number
set.seed(params$seed)
```

```
if (!dir.exists(output_dir)) dir.create(output_dir, recursive = TRUE)
```

Load GSEA results

```
read_gsea_results <- function(path_vector, ontology = "BP") {  
  ego_list <- lapply(path_vector, function(p) {  
    gsea <- readRDS(p)  
    gsea[[ontology]]@result  
  })  
  names(ego_list) <- names(path_vector)  
  return(ego_list)  
}  
  
if (!all(file.exists(gsea_paths))) {  
  stop("GSEA result file(s) not found: ", paste(gsea_paths[!file.exists(gsea_paths)],  
    ↪ collapse = ", "))  
}  
ego_list <- read_gsea_results(gsea_paths, ontology = "BP")
```

Build GO term × comparison matrix

```
padj_cutoff <- 0.05  
  
all_go_terms <- unique(unlist(  
  lapply(ego_list, function(x) {  
    sig <- x[x$p.adjust < padj_cutoff, ]  
    sig$ID  
  })  
)  
  
logp_mat <- matrix(0, nrow = length(all_go_terms), ncol = length(ego_list))  
rownames(logp_mat) <- all_go_terms  
  
bg_map <- msigdbr(species = "Homo sapiens", category = "C5", subcategory = "GO:BP") %>%  
  distinct(gs_name, gs_exact_source)  
  
all_go_terms <- as.data.frame(all_go_terms)  
colnames(all_go_terms) <- "Description"  
all_go_terms <- left_join(all_go_terms, bg_map, by = c("Description" = "gs_name"))  
colnames(logp_mat) <- names(ego_list)  
  
for (i in seq_along(ego_list)) {  
  ego_res <- ego_list[[i]]
```

```

ego_res <- ego_res[ego_res$p.adjust < padj_cutoff, ]
signed_logp <- -log10(ego_res$p.adjust) * sign(ego_res$NES)
names(signed_logp) <- ego_res$ID
logp_mat[names(signed_logp), i] <- signed_logp
}
rownames(logp_mat) <- all_go_terms$gs_exact_source

```

GO semantic similarity and clustering

```

go_sim <- godata(annoDb = "org.Hs.eg.db", ont = "BP", computeIC = TRUE)

##

## preparing gene to GO mapping data...

## preparing IC data...

go_terms <- rownames(logp_mat)
ic_values <- go_sim@IC[go_terms]
sim_matrix <- termSim(rownames(logp_mat), rownames(logp_mat), semData = go_sim, method =
  ↪ "Wang")
go_dist <- as.dist(1 - sim_matrix)

col_clust <- hclust(go_dist, method = "ward.D2")
col_dend <- as.dendrogram(col_clust)
clusters <- cutree(col_clust, k = k_num)

cluster_colors <- ggsci::pal_d3("category20")(k_num)
names(cluster_colors) <- as.character(1:k_num)
col_dend <- color_branches(col_dend, k = k_num, groupLabels = TRUE)

```

Heatmap annotation and color scale

```

top_annot <- HeatmapAnnotation(
  Cluster = as.factor(clusters),
  col = list(Cluster = cluster_colors),
  show_annotation_name = TRUE,
  annotation_name_side = "left"
)

col_fun <- colorRamp2(
  c(floor(min(logp_mat, na.rm = TRUE)), 0, ceiling(max(logp_mat, na.rm = TRUE))),
  c("#1B5B9D", "white", "#D7191C")
)

```

Draw heatmap and save (no display in report; only final ht used for files)

```
ht <- ComplexHeatmap::Heatmap(  
  t(logp_mat),  
  name = "-log10(p-value)",  
  top_annotation = top_annot,  
  cluster_columns = col_dend,  
  cluster_rows = FALSE,  
  col = col_fun,  
  border = TRUE,  
  show_row_names = TRUE,  
  show_column_names = FALSE,  
  column_names_gp = gpar(fontsize = 6),  
  column_names_rot = 45,  
  row_names_side = "left",  
  column_dend_height = unit(2, "cm"),  
  row_dend_width = unit(2, "cm"),  
  column_title = "GO Term Clusters by Semantic Similarity",  
  heatmap_legend_param = list(  
    title = bquote("Signed " * -log[10]("adj. p-value")),  
    at = seq(floor(min(logp_mat, na.rm = TRUE)), ceiling(max(logp_mat, na.rm = TRUE))), by  
    ↪ = 2)  
  )  
)  
  
# Reorder dendrogram colors to match drawn order  
ht_drawn <- draw(ht)
```

```
col_ordered <- column_order(ht_drawn)  
clustered_GO_terms <- rownames(logp_mat)[col_ordered]  
color_order <- unique(clusters[clustered_GO_terms])  
col_dend <- color_branches(col_dend, k = k_num, groupLabels = TRUE, col =  
  ↪ cluster_colors[color_order])  
  
ht <- ComplexHeatmap::Heatmap(  
  t(logp_mat),  
  name = "-log10(p-value)",  
  top_annotation = top_annot,  
  cluster_columns = col_dend,  
  cluster_rows = FALSE,  
  col = col_fun,  
  border = TRUE,  
  show_row_names = TRUE,  
  show_column_names = FALSE,  
  column_names_gp = gpar(fontsize = 6),  
  column_names_rot = 45,  
  row_names_side = "left",  
  column_dend_height = unit(2, "cm"),
```

```

row_dend_width = unit(2, "cm"),
column_title = "GO Term Clusters by Semantic Similarity",
heatmap_legend_param = list(
  title = bquote("Signed " * -log[10]("adj. p-value")),
  at = seq(floor(min(logp_mat, na.rm = TRUE)), ceiling(max(logp_mat, na.rm = TRUE)), by
    ↪ = 2)
)
)
)

```

```

eps_path <- file.path(output_dir, "heatmap_plot.eps")
cairo_ps(
  filename = eps_path,
  width = max(ceiling(dim(logp_mat)[1] * 0.06), 6),
  height = max(ceiling(dim(logp_mat)[2] * 0.5), 4),
  family = "Arial",
  onefile = FALSE,
  fallback_resolution = 600
)
draw(ht)
dev.off()

```

```

## pdf
## 2

```

```

pdf_path <- file.path(output_dir, "heatmap_plot.pdf")
cairo_pdf(
  filename = pdf_path,
  width = max(ceiling(dim(logp_mat)[1] * 0.06), 6),
  height = max(ceiling(dim(logp_mat)[2] * 0.5), 4),
  family = "Arial",
  fallback_resolution = 600
)
draw(ht)
dev.off()

```

```

## pdf
## 2

```

```

message("Saved: ", eps_path, " and ", pdf_path)

```

```

## Saved: /Users/xliu2942/Documents/Projects/MAXOMOD/MAXOMOD_CSF/Plots/Fig1f_GESA_IC/heatmap_plot.eps and

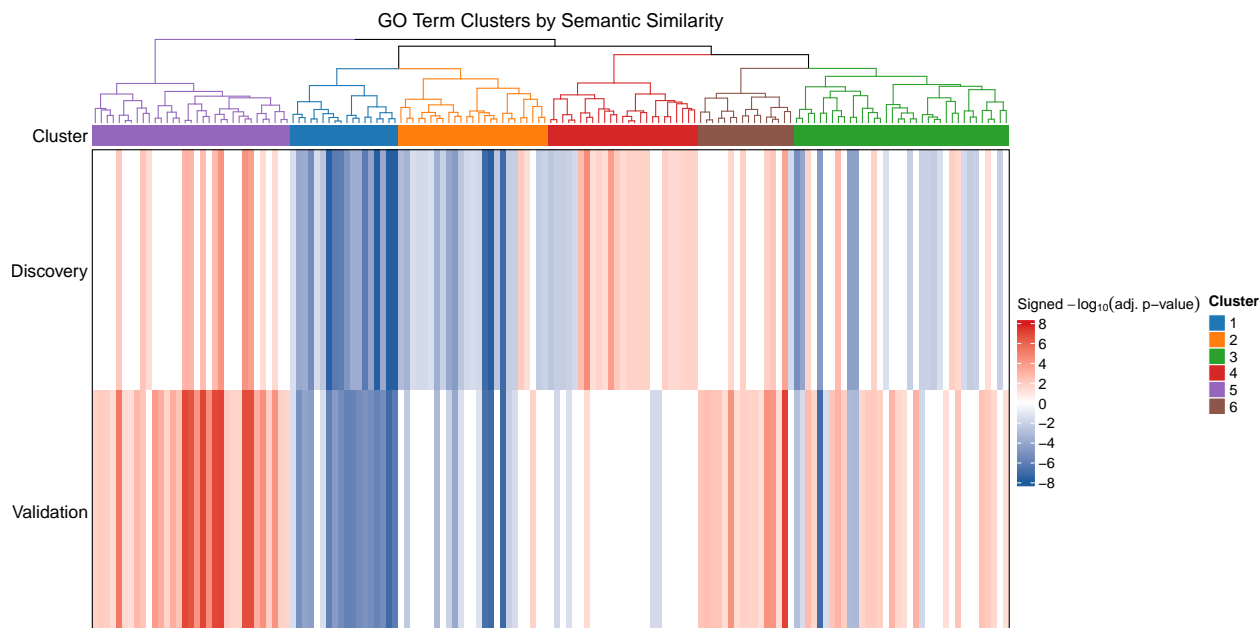
```

Preview in HTML

```

draw(ht)

```



Export summary CSV

```
term_info <- ego_list[[1]]
all_term_info <- lapply(ego_list, function(x) x[, c("ID", "Description")]) %>%
  bind_rows() %>%
  distinct(ID, .keep_all = TRUE) %>%
  left_join(all_go_terms, by = c("Description" = "Description")) %>%
  dplyr::select(gs_exact_source, Description) %>%
  rename(ID = gs_exact_source)
term_desc_full_map <- setNames(all_term_info$Description, all_term_info$ID)

df_long <- as.data.frame(logp_mat) %>%
  tibble::rownames_to_column("GO_term") %>%
  pivot_longer(-GO_term, names_to = "Comparison", values_to = "log10_pvalue")
df_long$Cluster <- clusters[df_long$GO_term]
df_long_filtered <- df_long %>%
  mutate(p_value = 10^(-log10_pvalue),
         IC = ic_values[GO_term],
         description = term_desc_full_map[GO_term])

go_representatives <- df_long_filtered %>%
  group_by(Cluster) %>%
  slice_min(IC, n = 1, with_ties = FALSE) %>%
  ungroup() %>%
  mutate(description = term_desc_full_map[GO_term])

df_long_filtered <- df_long_filtered %>%
  left_join(
    go_representatives %>% dplyr::select(Cluster, Cluster_summary = description),
```

```

    by = "Cluster"
  )
df_long_filtered$description <- gsub("GOBP_", "", df_long_filtered$description)
df_long_filtered$Cluster_summary <- gsub("GOBP_", "", df_long_filtered$Cluster_summary)

output_csv_path <- file.path(output_dir, "GSEA_results_summary.csv")
write.csv(df_long_filtered, output_csv_path, row.names = FALSE)
message("Exported: ", output_csv_path)

```

Exported: /Users/xliu2942/Documents/Projects/MAXOMOD/MAXOMOD-CSF/Plots/Fig1f_GESA_IC/GSEA_results_summary.csv

Outputs

- **Heatmap:** heatmap_plot.eps and heatmap_plot.pdf in /Users/xliu2942/Documents/Projects/MAXOMOD/MAXOMOD-CSF/Plots/Fig1f_GESA_IC/
- **Summary table:** GSEA_results_summary.csv in /Users/xliu2942/Documents/Projects/MAXOMOD/MAXOMOD-CSF/Plots/Fig1f_GESA_IC/

Inputs are read from **demo/Discovery/06_GSEA/GSEA_result.rds** and **demo/Validation/06_GSEA/GSEA_result.rds** (already copied into demo folder).