# =====================================================

# 🚀 模块 GO 富集分析与可视化（适用于任意模块）

# =====================================================

suppressPackageStartupMessages({

library(clusterProfiler)

library(tidyverse)

library(readxl)

library(ggh4x)

})

# =====================================================

# 1️⃣ 路径设置

# =====================================================

go\_annotation\_path <- "/home/data/t210549/liuhuacheng/wrky注释/全基因组/extracted\_gene\_go\_id.csv"

go\_annotation\_full\_path <- "/home/data/t210549/liuhuacheng/wrky注释/全基因组/all\_go\_long.csv"

module\_gene\_path <- "/home/data/t210549/liuhuacheng/wrky注释/全基因组/性状相关模块/GO/Module\_cyan\_gene.xlsx"

out\_prefix <- "/home/data/t210549/liuhuacheng/wrky注释/全基因组/性状相关模块/GO/Module\_cyan\_GO"

# =====================================================

# 2️⃣ 读取 GO 注释表

# =====================================================

go\_annotation <- read.csv(go\_annotation\_path, header = FALSE, stringsAsFactors = FALSE)

colnames(go\_annotation) <- c("GO\_ID", "Entrez\_ID")

cat(sprintf("[INFO] ✅ GO 注释加载完成：%d 条映射，涉及 %d 个 GO 号。\n",

nrow(go\_annotation), length(unique(go\_annotation$GO\_ID))))

# =====================================================

# 3️⃣ 读取模块基因

# =====================================================

if (!file.exists(module\_gene\_path)) stop("❌ 未找到模块基因文件：", module\_gene\_path)

module\_genes <- read\_excel(module\_gene\_path) %>%

pull(GENEID) %>% as.character() %>% unique() %>% na.omit()

if (length(module\_genes) == 0) stop("❌ 模块基因列表为空，请检查 Excel 文件列名是否为 GENEID！")

cat(sprintf("[INFO] 模块基因数：%d 个。\n", length(module\_genes)))

# =====================================================

# 4️⃣ 进行 GO 富集分析

# =====================================================

ego <- enricher(

gene = module\_genes,

TERM2GENE = go\_annotation,

pvalueCutoff = 0.2,

qvalueCutoff = 0.5

)

if (is.null(ego) || nrow(as.data.frame(ego)) == 0) {

stop("❌ 当前阈值下无显著 GO 富集结果，请尝试放宽 pvalueCutoff 或检查注释文件。")

}

ego\_df <- as.data.frame(ego)

cat(sprintf("[INFO] 富集结果：%d 条显著 GO 条目。\n", nrow(ego\_df)))

# =====================================================

# 5️⃣ 合并 GO 术语与 Ontology 分类

# =====================================================

go\_full <- read.csv(go\_annotation\_full\_path, header = TRUE, stringsAsFactors = FALSE)

if (!all(c("go\_id", "term", "ontology") %in% colnames(go\_full))) {

stop("❌ all\_go\_long.csv 文件缺少必要列：go\_id, term, ontology。")

}

ego\_df <- ego\_df %>%

mutate(ID\_clean = toupper(gsub("^GO:", "", ID))) %>%

left\_join(

go\_full %>%

mutate(go\_id\_clean = toupper(gsub("^GO:", "", go\_id))) %>%

select(go\_id\_clean, term, ontology),

by = c("ID\_clean" = "go\_id\_clean")

)

matched <- sum(!is.na(ego\_df$term))

cat(sprintf("[INFO] 成功匹配 GO 描述：%d/%d (%.1f%%)\n",

matched, nrow(ego\_df), 100 \* matched / nrow(ego\_df)))

ego\_df <- ego\_df %>% filter(!is.na(term) & !is.na(ontology))

# =====================================================

# 6️⃣ 修复 Count 列（若缺失则重新计算）

# =====================================================

if (!"Count" %in% colnames(ego\_df) || all(is.na(ego\_df$Count))) {

cat("[WARN] 未找到 Count 列，正在从 geneID 重新计算...\n")

ego\_df$Count <- sapply(strsplit(ego\_df$geneID, "/"), length)

}

ego\_df$Count <- as.numeric(ego\_df$Count)

# =====================================================

# 7️⃣ 每个 Ontology 取前 10 条（按 Count 降序）

# =====================================================

Go\_count <- ego\_df %>%

distinct(term, ontology, .keep\_all = TRUE) %>%

select(GO\_term = term, Aspect = ontology, Count) %>%

mutate(Aspect = factor(Aspect, levels = c("BP", "MF", "CC"))) %>%

group\_by(Aspect) %>%

arrange(desc(Count)) %>%

slice\_head(n = 10) %>%

ungroup() %>%

arrange(Aspect, desc(Count))

if (nrow(Go\_count) == 0) stop("❌ 没有可用于绘图的 GO 条目。请检查 ontology 是否匹配。")

cat(sprintf("[INFO] 用于绘图的 GO 条目：%d 条。\n", nrow(Go\_count)))

print(table(Go\_count$Aspect))

# =====================================================

# 8️⃣ 动态计算 X 轴范围

# =====================================================

max\_count <- max(Go\_count$Count, na.rm = TRUE)

x\_max <- max\_count \* 1.2

cat(sprintf("[INFO] Count 最大值：%d，X 轴上限：%d\n", max\_count, x\_max))

# =====================================================

# 9️⃣ 绘图数据处理

# =====================================================

Go\_count\_plot <- Go\_count %>%

mutate(GO\_term\_wrapped = str\_wrap(GO\_term, width = 40),

display\_label = factor(GO\_term\_wrapped, levels = rev(unique(GO\_term\_wrapped))))

# =====================================================

# 🔟 绘图（分面 + 自定义主题）

# =====================================================

plot\_final <- ggplot(Go\_count\_plot, aes(x = Count, y = display\_label, fill = Aspect)) +

geom\_col(width = 0.8) +

geom\_text(aes(label = Count), hjust = -0.2, size = 3.5, fontface = "bold") +

scale\_x\_continuous(expand = expansion(mult = c(0, 0.15)), limits = c(0, x\_max)) +

scale\_fill\_manual(values = c("BP" = "#74c476", "MF" = "#41b6c4", "CC" = "#9e9ac8")) +

facet\_grid(Aspect ~ ., scales = "free\_y", space = "free\_y") +

ggtitle("GO Enrichment: Module Cyan") +

labs(x = "Count", y = NULL) +

theme\_bw(base\_size = 12) +

theme(

strip.background = element\_rect(fill = "white", color = "black"),

strip.text.y = element\_text(size = 12, face = "bold", color = "black"),

plot.title = element\_text(size = 16, face = "bold", hjust = 0.5),

axis.text.x = element\_text(size = 10, face = "bold"),

axis.text.y = element\_text(size = 9, color = "black"),

axis.title.x = element\_text(size = 12, face = "bold"),

legend.position = "none",

panel.border = element\_rect(linewidth = 0.8, color = "black"),

panel.grid.major.x = element\_line(color = "grey90", linewidth = 0.3),

panel.grid.major.y = element\_blank(),

plot.margin = margin(t = 0.5, r = 1, b = 0.5, l = 1.5, "cm")

)

# =====================================================

# 1️⃣1️⃣ 保存 PDF 图

# =====================================================

plot\_height <- max(8, nrow(Go\_count\_plot) \* 0.4)

dir.create(dirname(out\_prefix), recursive = TRUE, showWarnings = FALSE)

ggsave(paste0(out\_prefix, "\_BarPlot.pdf"), plot\_final, width = 12, height = plot\_height, dpi = 300)

cat("[DONE] ✅ 模块 GO 富集分析与可视化完成！\n",

"📁 结果文件：", paste0(out\_prefix, "\_BarPlot.pdf"), "\n")