

Differential Expression Analysis for bulk RNA-seq data

CTRL Condition: Vehicle 100 vs 0

Ximing Ran

2025-03-24

Contents

1. Read the count data	2
2. Differential expression analysis	2
3. Visualization for result	4
(1) Sample information	4
(2) DEG visualization - Volcano plot and Heatmap	6
4. GSEA analysis	15
5. Pathway Enrichment Analysis	18
Session information	30

```
library(tibble)
library(tidyr)
library(dplyr)
library(rtracklayer)
```

```
# load function from local files
source(here::here("source", "DEG_functions.R"))
```

1. Read the count data

In this section, we will read the clean count data from the `synaptosomes_bulkRNA` folder. We will read the data and merge them into a single table. The final table will be stored in `../dataresults/bulkRNA_counts_clean.csv`.

```
input_count <- read.csv(here::here("data", "bulkRNA",
                                   "bulkRNA_counts_cleaned.csv"))

counts <- as.data.frame(input_count) %>%
  column_to_rownames(var = "gene")
colnames(counts) <- gsub("_", "-", colnames(counts))

# raw sample list
sample_list_raw <- read.csv(here::here("data", "bulkRNA",
                                       "sample_info_AD.csv")) %>%
  mutate(condition = paste0(Diagnosis, "_", Treatment),
         sample = gsub("_", "-", Sample.name))

# Ensure the column names of counts exist in Sample.name
new_colnames <- sample_list_raw$Label[match(colnames(counts), sample_list_raw$sample)]

# Assign new column names
colnames(counts) <- new_colnames

# sort the columns by the colname
condition_list <- data.frame(
  group = sample_list_raw$condition
)

row.names(condition_list) <- sample_list_raw$Label

counts <- counts[, rownames(condition_list)]

gene_name_mapping <- readRDS(here::here("data", "ref", "gene_name_mapping.rds"))
```

2. Differential expression analysis

In this section, we will perform differential expression analysis using DESeq2. We will compare the 22q vs Control in the vehicle condition. The results will be stored in `results/02-DEG-V_10/DESeq2_results.csv`.

```
# Init the result folder structure for the result
result_folder_all = './results/02-DEG-AD_0_vs_100'
result_folder = result_folder_all
Result_folder_structure(result_folder)

# load the comparison group information
reference_group <- "AD_Veh_0"
compare_group <- "AD_Veh_100"
reference_group_short <- reference_group
compare_group_short <- compare_group
```

```

filter_sample_info <- condition_list %>%
  filter(group %in% c(reference_group, compare_group))
filter_counts <- counts[, rownames(filter_sample_info)]

# Run the DESeq2 analysis
dds_obj <- DESeqAnalysis(counts = filter_counts,
  reference_group = reference_group,
  compare_group = compare_group,
  condition_list = filter_sample_info,
  target_gene = target_gene,
  result_folder = result_folder)

res <- results(dds_obj)
resOrdered <- res[order(res$padj), ]

# omit the NA values
resOrdered <- resOrdered[!is.na(resOrdered$padj),]
dds_obj <- dds_obj[rownames(resOrdered),]
write.csv(resOrdered, file.path(result_folder, "02-DEG", "01_all_gene_results.csv"))

# DEG with log2fc > 1 and padj < 0.05
deg_1 <- resOrdered %>% as.data.frame() %>% rownames_to_column(var = "gene") %>%
  filter(padj < 0.05 & abs(log2FoldChange) > 1) %>% arrange(padj)
deg_1 <- deg_1[!is.na(deg_1$padj),]
write.csv(deg_1, file.path(result_folder, "02-DEG", "02_DEG_log2fc_1.csv"), row.names = FALSE)

# DEG with log2fc > 1.5 and padj < 0.05
deg_1.5 <- resOrdered %>% as.data.frame() %>% rownames_to_column(var = "gene") %>%
  filter(padj < 0.05 & abs(log2FoldChange) > 1.5) %>% arrange(padj)
deg_1.5 <- deg_1.5 [!is.na(deg_1.5 $padj),]
write.csv(deg_1.5 , file.path(result_folder, "02-DEG", "03_DEG_log2fc_1_5.csv"), row.names = FALSE)
print("DEG analysis is done")

## [1] "DEG analysis is done"

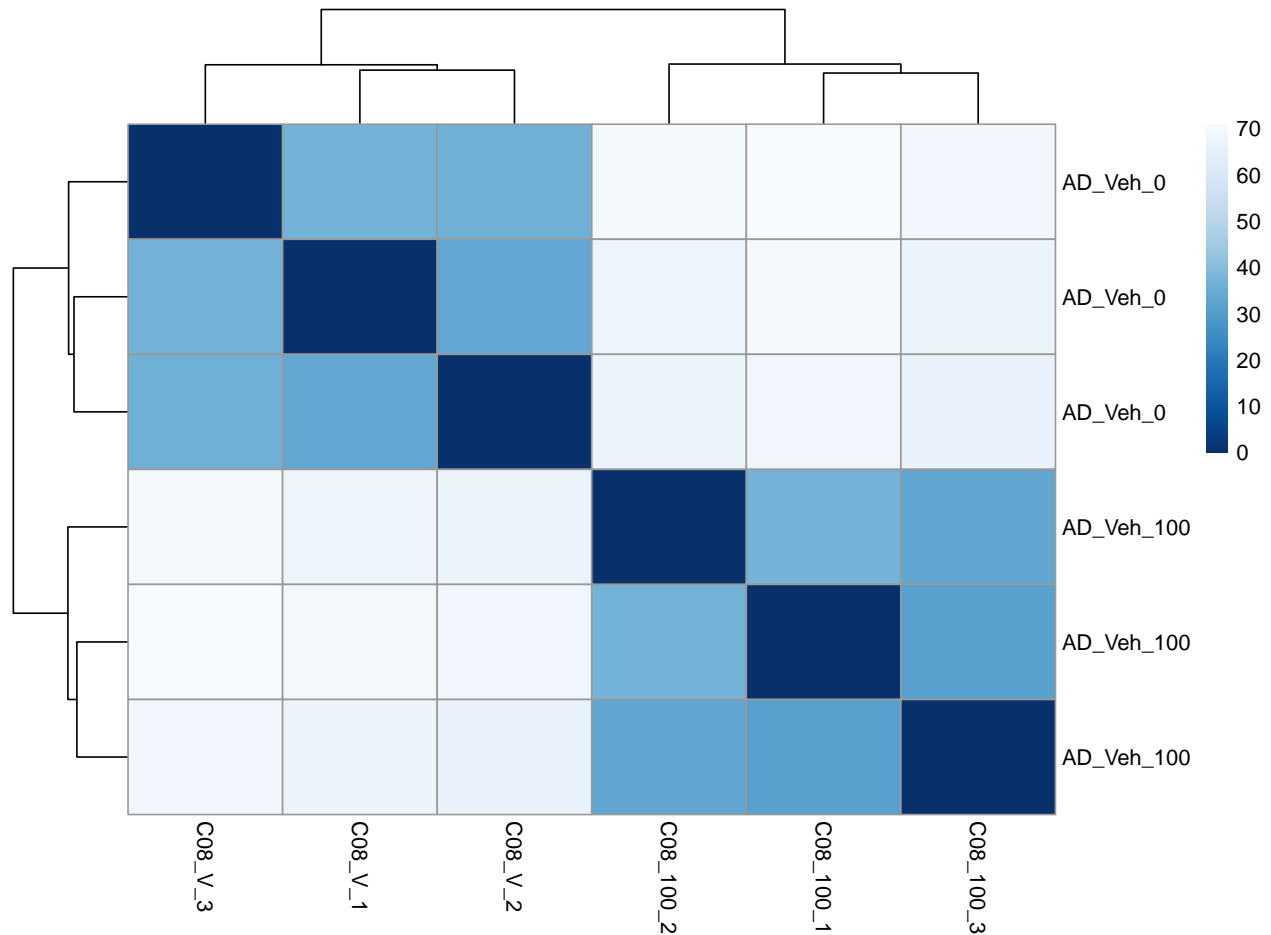
# Save the normalized counts
normalized_counts <- counts(dds_obj, normalized = TRUE)
write.csv(normalized_counts, file.path(result_folder, "02-DEG", "DESeq2_normalized_counts.csv"))

```

3. Visualization for result

(1) Sample information

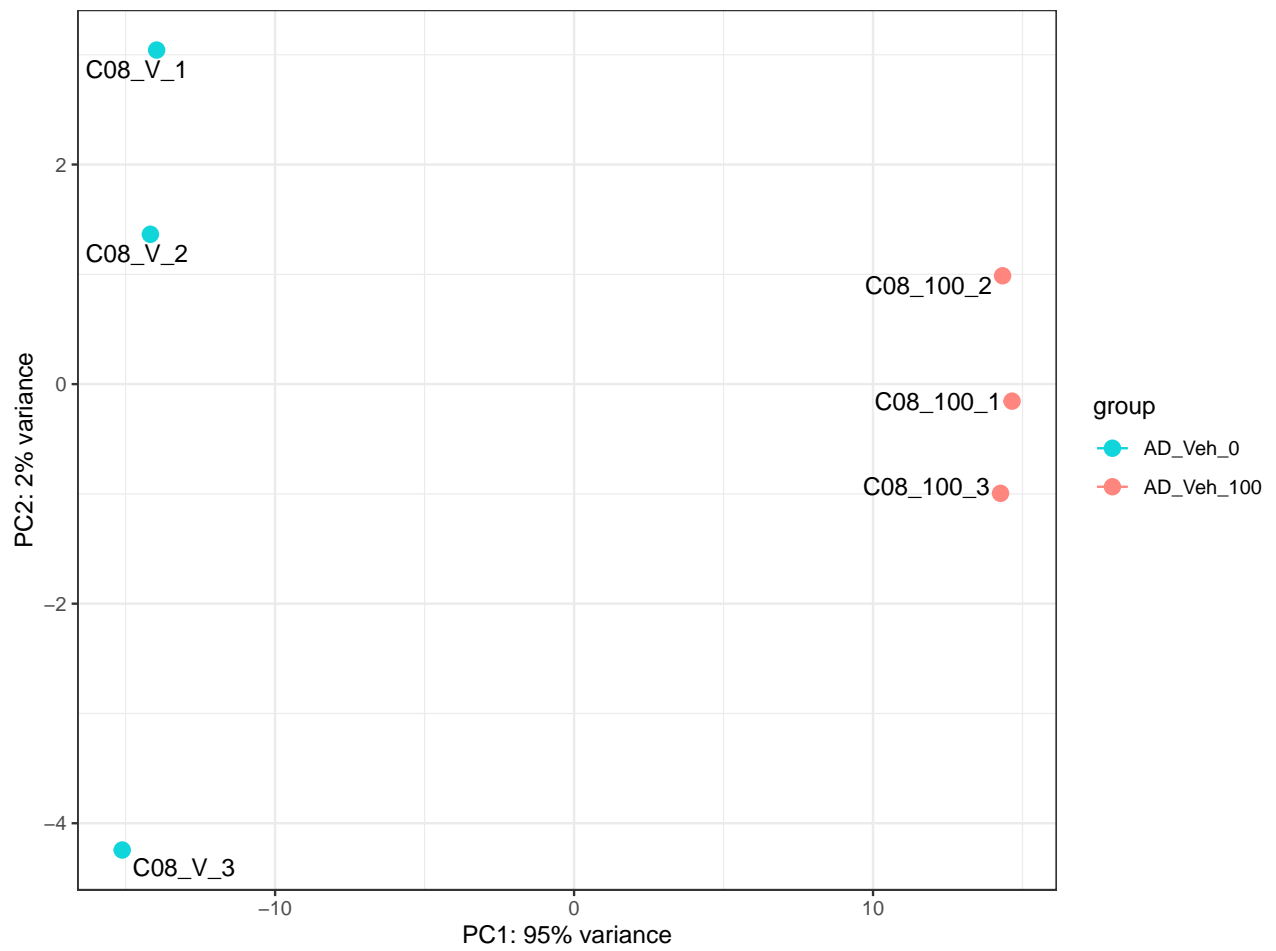
```
# Plot the sample distance  
plot_sample_heatmap(dds_obj, figure_folder =  
    file.path(result_folder, "01-Sample_info"),  
    file_name = "01_sample_distance_heatmap")
```



```
## [1] "Sample distance heatmap is done"
```

```
# Plot the PCA plot for the sample  
plot_sample_PCA_plot(dds_obj, figure_folder = file.path(result_folder, "01-Sample_info"),  
    file_name = "02_sample_PCA_plot",  
    reference_group, compare_group)
```

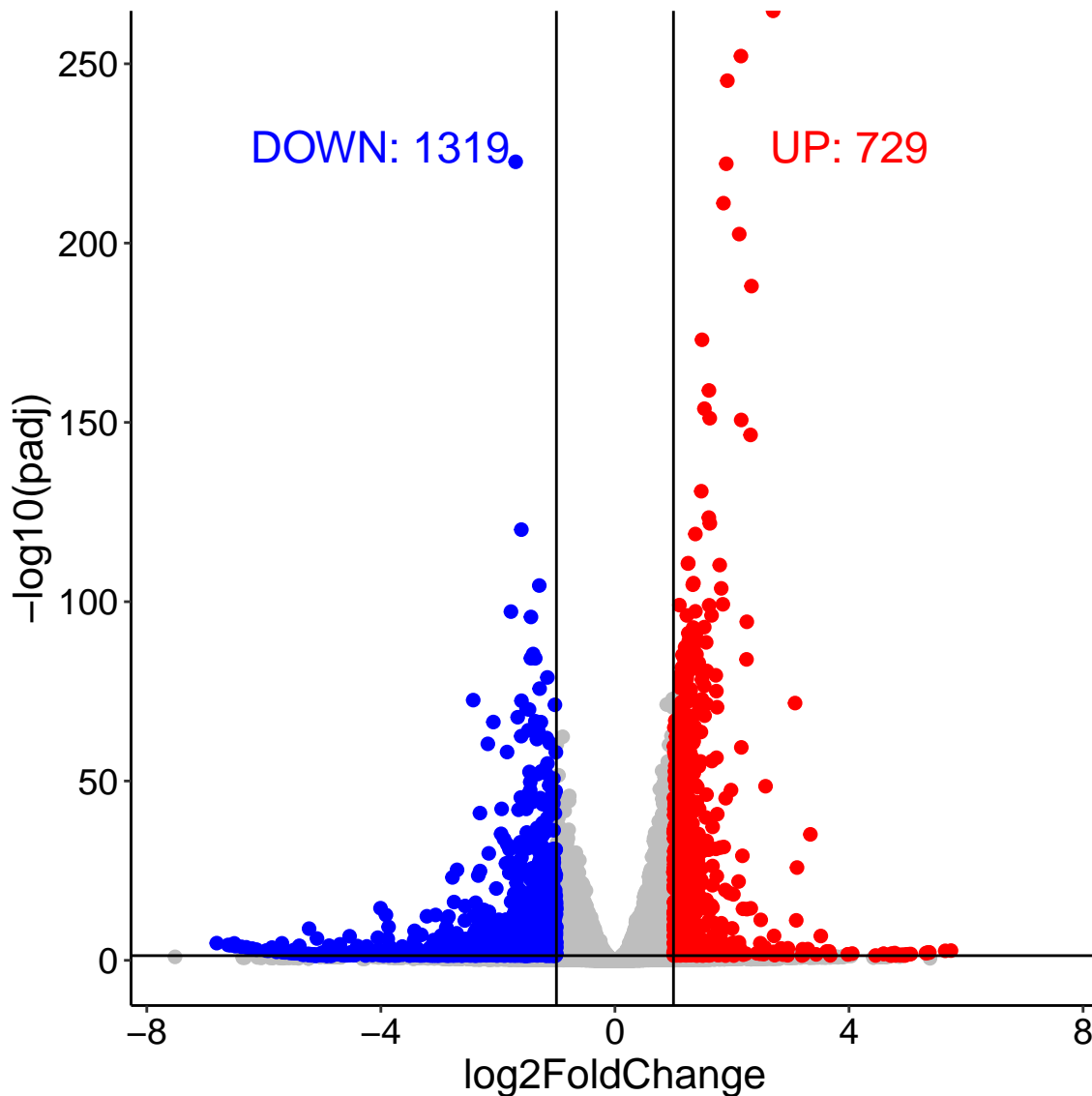
```
## [1] "PCA plot is done"
```



(2) DEG visualization - Volcano plot and Heatmap

```
result_df <- results(dds_obj) %>%  
  as.data.frame() %>%  
  rownames_to_column(var = "GeneName") %>%  
  dplyr::select(GeneName, everything()) %>%  
  filter(!is.na(padj)) %>% # Correct way to filter non-NA values  
  arrange(padj)  
  
# Plot the volcano plot for the DEG  
plot_volcano_plot(result_df=result_df,  
  figure_folder = file.path(result_folder,"02-DEG"),  
  file_name = "02_volcano_plot_log2fc_1",  
  thread = 1 , dot_size =2, label_gene = NULL)
```

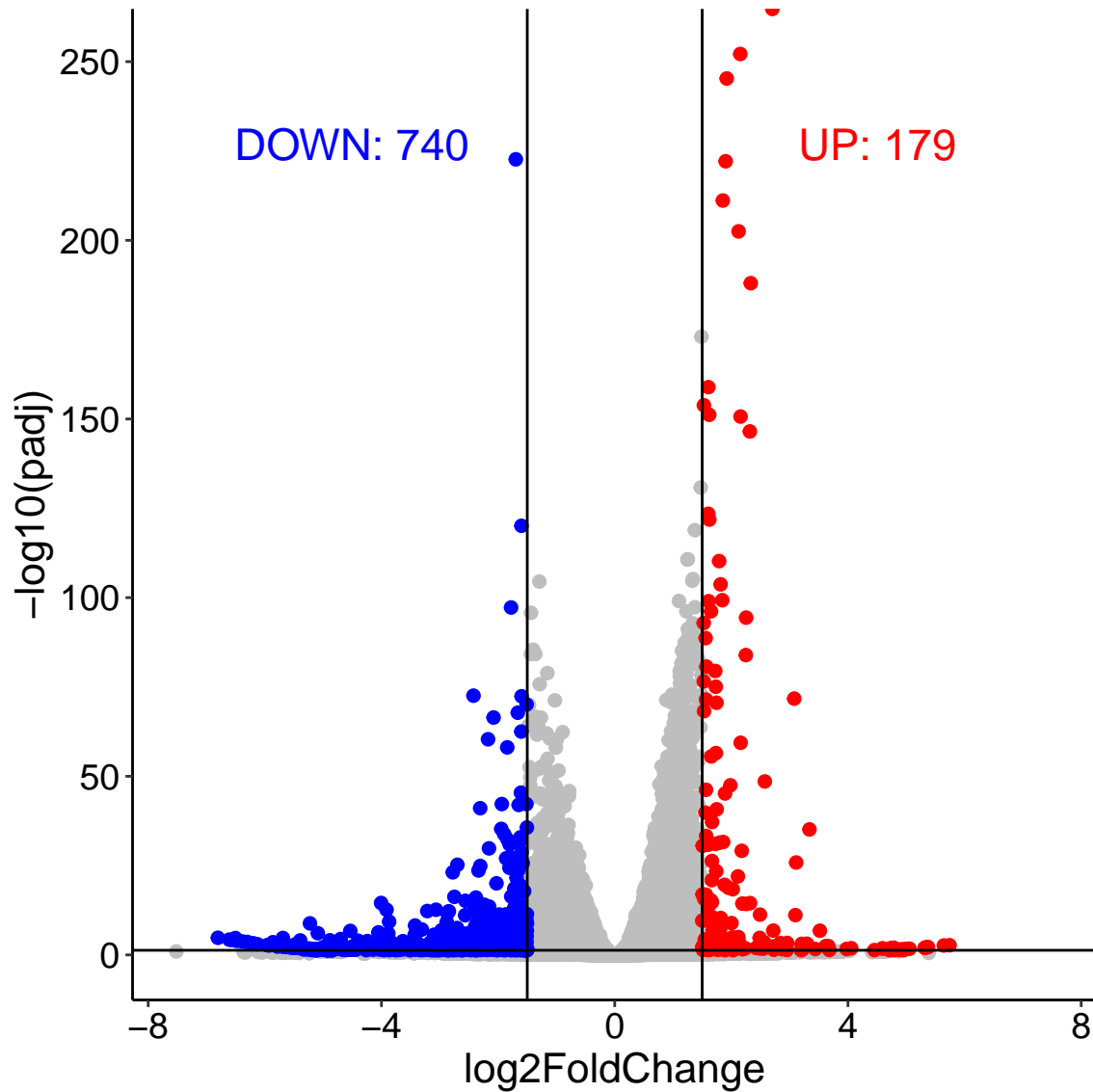
```
## [1] "Volcano plot for 02_volcano_plot_log2fc_1"
```



```
plot_volcano_plot(result_df=result_df,
```

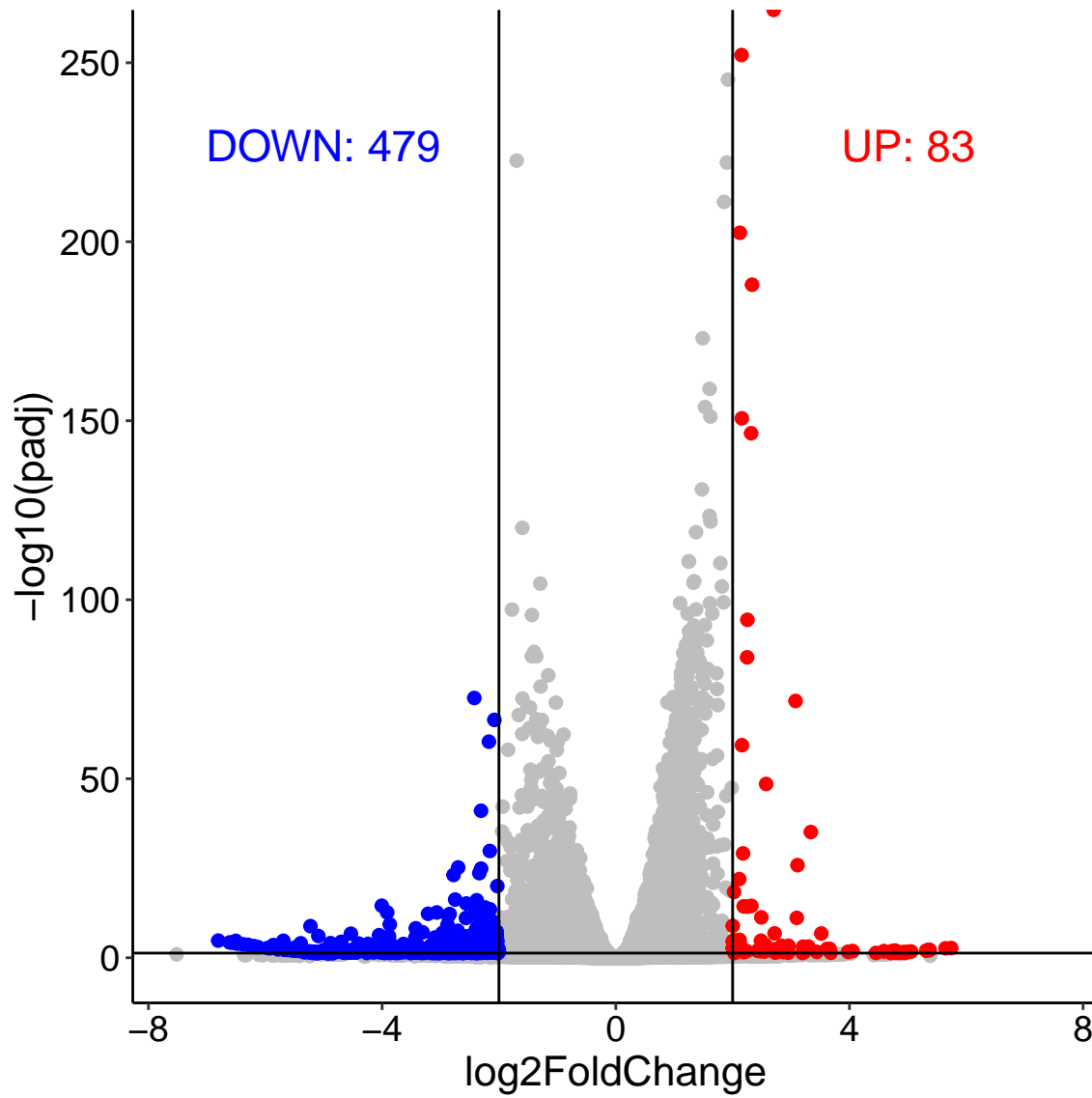
```
figure_folder = file.path(result_folder,"02-DEG"),
file_name = "03_volcano_plot_log2fc_1.5",
thread = 1.5 , dot_size =2,label_gene = NULL)
```

```
## [1] "Volcano plot for 03_volcano_plot_log2fc_1.5"
```



```
plot_volcano_plot(result_df=result_df,
figure_folder = file.path(result_folder,"02-DEG"),
file_name = "03_volcano_plot_log2fc_2",
thread = 2 , dot_size =2,label_gene = NULL)
```

```
## [1] "Volcano plot for 03_volcano_plot_log2fc_2"
```

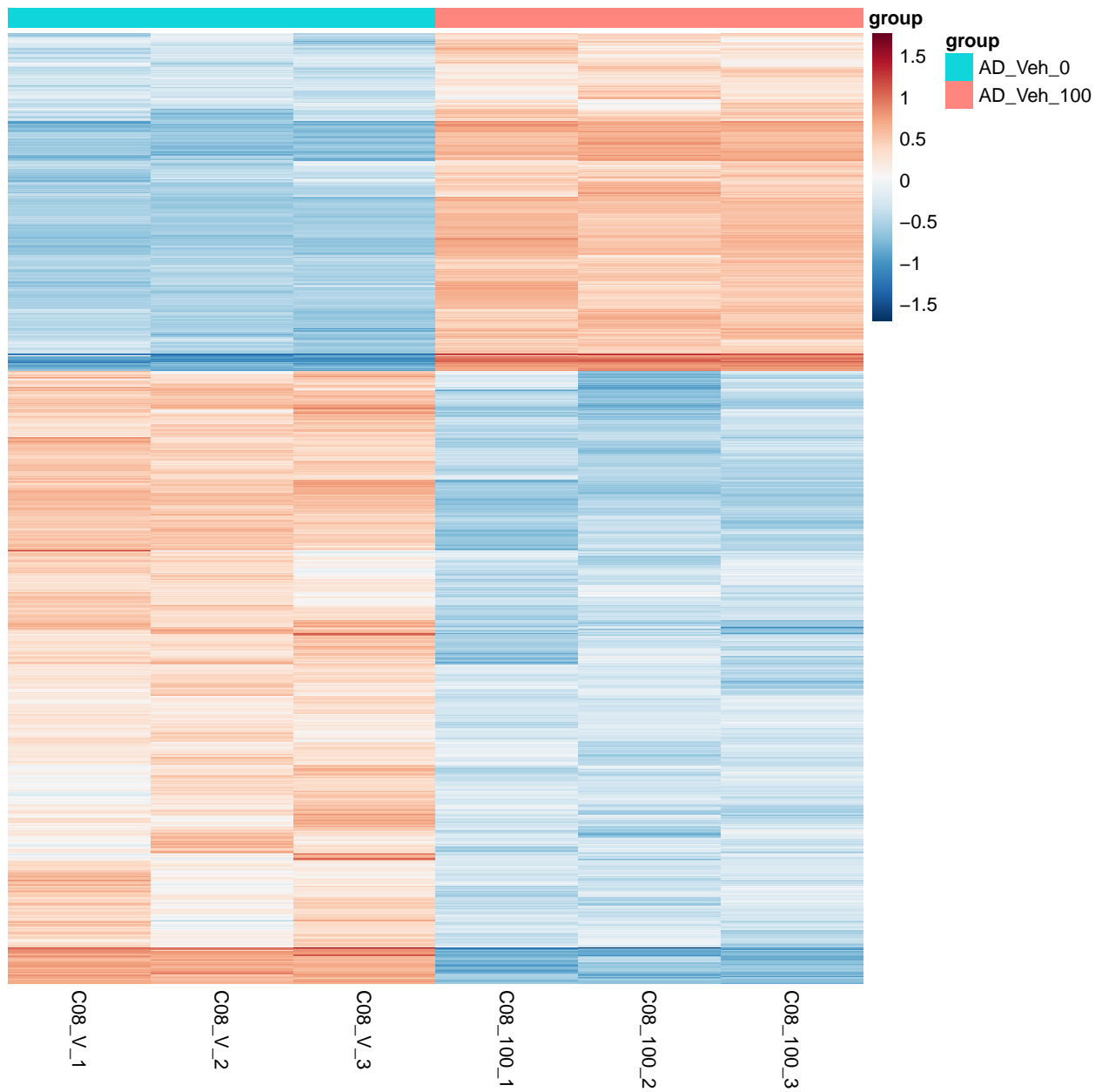


```
# Plot the heatmap for the DEG
vsd_obj <- varianceStabilizingTransformation(dds_obj, blind = TRUE)

DEG_gene_1 <- result_df %>% filter(abs(log2FoldChange) > 1 & padj < 0.05) %>% pull(GeneName)
DEG_gene_1.5 <- result_df %>% filter(abs(log2FoldChange) > 1.5 & padj < 0.05) %>% pull(GeneName)
DEG_gene_2 <- result_df %>% filter(abs(log2FoldChange) > 2 & padj < 0.05) %>% pull(GeneName)

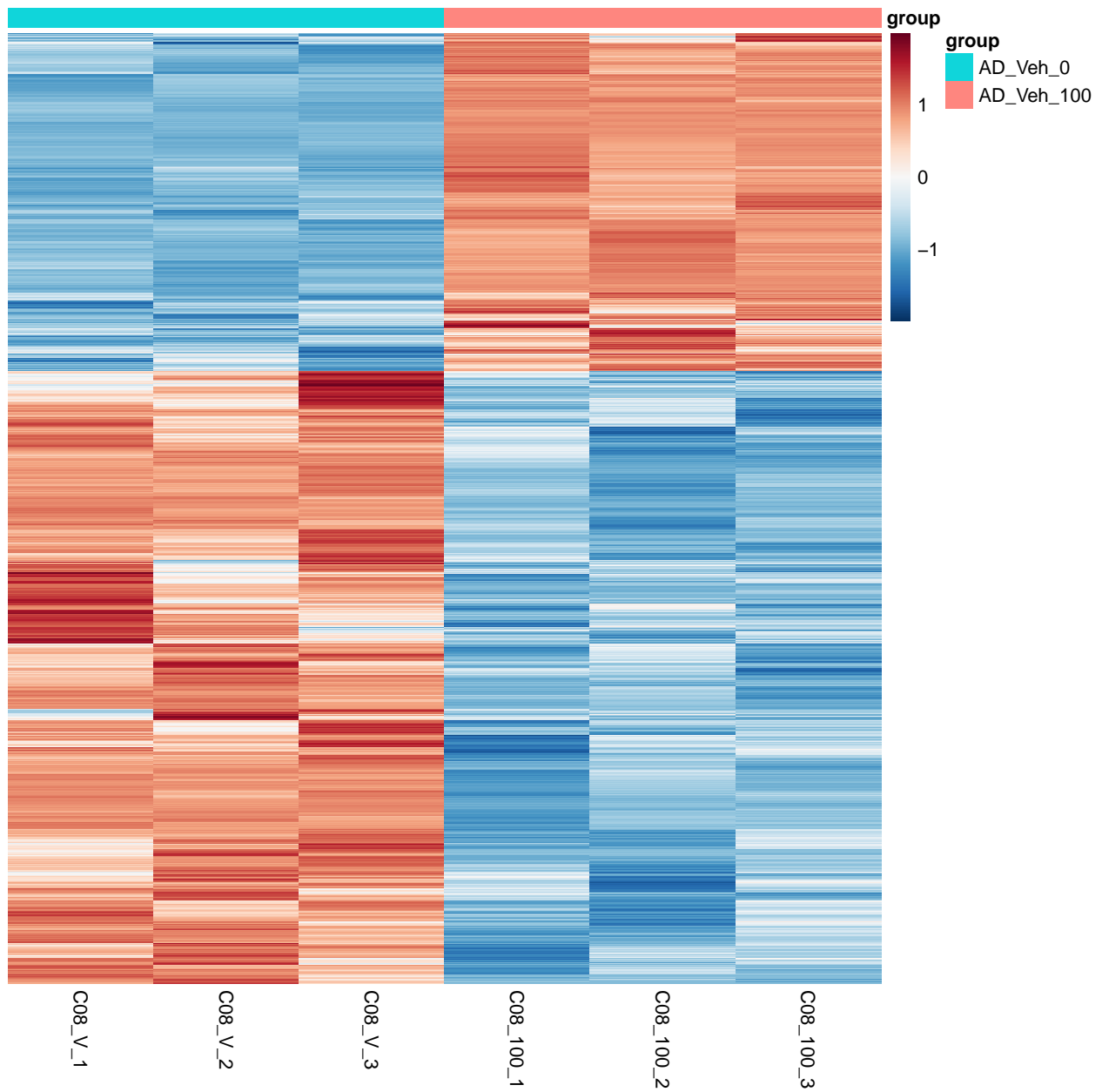
plot_gene_heatmap(vsd_obj, gene_list = DEG_gene_1,
  figure_folder = file.path(result_folder, "O2-DEG"),
  file_name = "O2_heatmap_log2fc_1",
  reference_group, compare_group,
  cluster_rows = TRUE, cluster_cols = FALSE,
  scale = "none")

## [1] "Heatmap for O2_heatmap_log2fc_1 "
```

```
plot_gene_heatmap(vsd_obj, gene_list = DEG_gene_1,
                  figure_folder = file.path(result_folder, "02-DEG"),
                  file_name = "02_heatmap_log2fc_1_row",
                  reference_group, compare_group,
                  cluster_rows = TRUE, cluster_cols = FALSE,
                  scale = "row")
```

```
## [1] "Heatmap for 02_heatmap_log2fc_1_row "
```



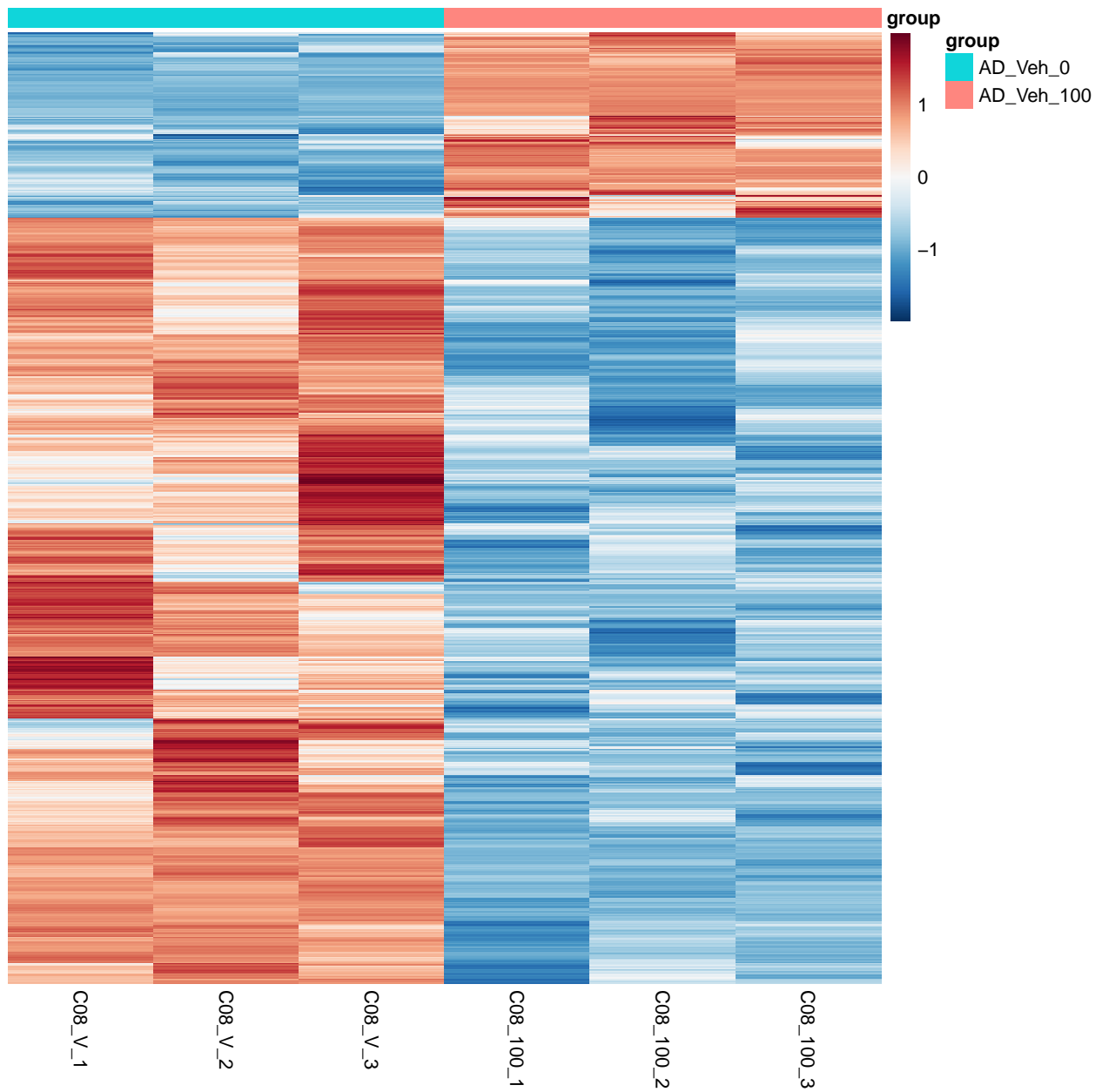
```
plot_gene_heatmap(vsd_obj, gene_list = DEG_gene_1.5,
                  figure_folder = file.path(result_folder, "02-DEG"),
                  file_name = "03_heatmap_log2fc_1.5",
                  reference_group, compare_group,
                  cluster_rows = TRUE, cluster_cols = FALSE,
                  scale = "none")
```

```
## [1] "Heatmap for 03_heatmap_log2fc_1.5 "
```



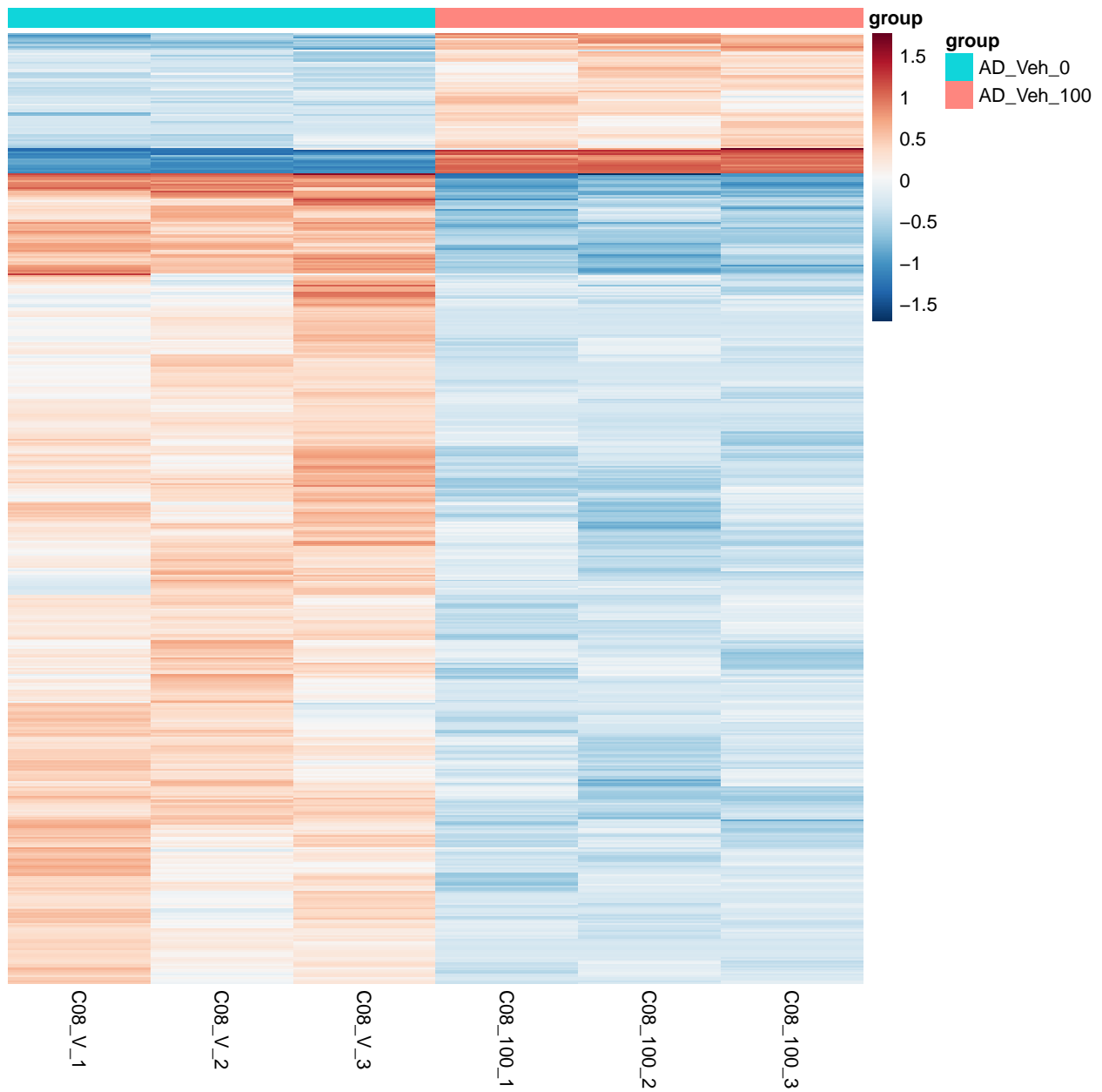
```
plot_gene_heatmap(vsd_obj, gene_list = DEG_gene_1.5,
                  figure_folder = file.path(result_folder, "02-DEG"),
                  file_name = "03_heatmap_log2fc_1.5_row",
                  reference_group, compare_group,
                  cluster_rows = TRUE, cluster_cols = FALSE,
                  scale = "row")
```

```
## [1] "Heatmap for 03_heatmap_log2fc_1.5_row "
```



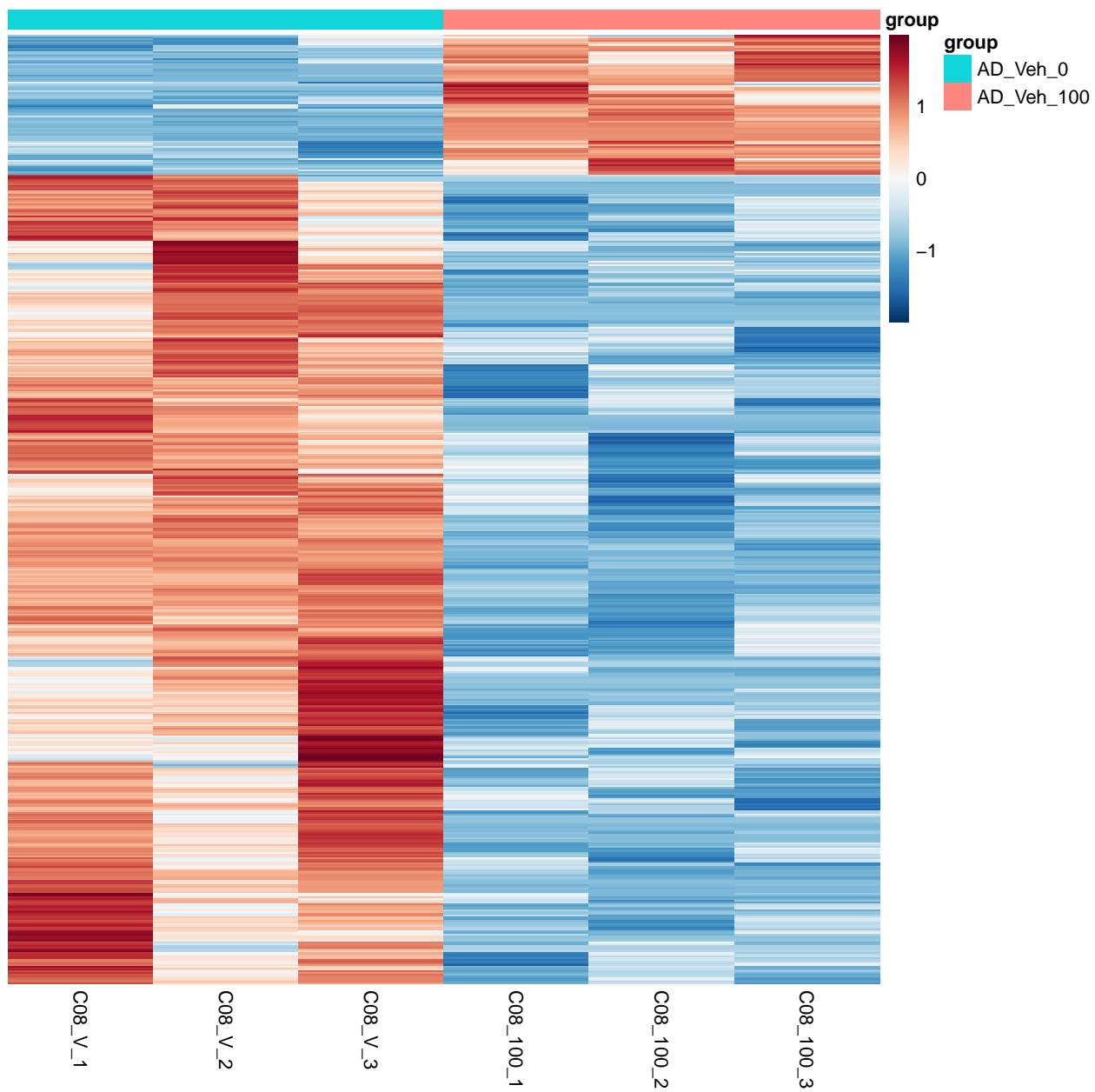
```
plot_gene_heatmap(vsd_obj, gene_list = DEG_gene_2,
                  figure_folder = file.path(result_folder, "02-DEG"),
                  file_name = "03_heatmap_log2fc_2",
                  reference_group, compare_group,
                  cluster_rows = TRUE, cluster_cols = FALSE,
                  scale = "none")
```

```
## [1] "Heatmap for 03_heatmap_log2fc_2 "
```



```
plot_gene_heatmap(vsd_obj, gene_list = DEG_gene_2,
                  figure_folder = file.path(result_folder, "02-DEG"),
                  file_name = "03_heatmap_log2fc_2_row",
                  reference_group, compare_group,
                  cluster_rows = TRUE, cluster_cols = FALSE,
                  scale = "row")
```

```
## [1] "Heatmap for 03_heatmap_log2fc_2_row "
```



4. GSVA analysis

```
# The following code is used to generate the GSVA matrix , only need to run once
gmxFFile <- here::here("data", "ref", "c5.go.v2023.1.Hs.symbols.gmt")
go_list <- getGmt(gmxFFile)

geneset <- go_list
dat <- as.matrix(counts)

gsvaparam <- gsvaParam(dat, geneset, maxDiff=TRUE)
gsva_es <- gsva(gsvaparam)

## No annotation package name available in the input data object.
## Attempting to directly match identifiers in data to gene sets.

## Estimating GSVA scores for 10531 gene sets.
## Estimating ECDFs with Gaussian kernels
## |

gsva_matrix <- as.data.frame(gsva_es)

# save the result
write.csv(gsva_matrix, file.path(result_folder, "04-GSVA", "01_GSVA_matrix.csv"))

gsva_matrix <- read.csv(file.path(result_folder, "04-GSVA", "01_GSVA_matrix.csv"),
  row.names = 1)

colnames(gsva_matrix) <- sub("^X", "", colnames(gsva_matrix))
condition_list_label <- condition_list %>%
  filter(group %in% c(reference_group, compare_group)) %>%
  mutate(group = case_when(
    group == reference_group ~ reference_group_short,
    group == compare_group ~ compare_group_short,
    TRUE ~ group # fallback just in case
  ))

# plot the heatmap for the GSVA result
pathway_list <- read.csv(here::here("data", "ref", "focus-pathway_2024_10_03.csv"))

# # plot for all pathway
# for (i in 1:nrow(pathway_list)){
#   if (i %% 10 == 0) print(i)
#   pathway_name <- pathway_list$pathway[i]
#   plot_gsva_boxplot(gsva_matrix,
#     condition_list_label = condition_list_label,
#     pathway_name = pathway_name,
#     figure_folder = file.path(result_folder, "04-GSVA", "Boxplot"),
#     file_name = paste0("GSVA_", pathway_name),
#     fig.height = 6, fig.width = 4,
#     reference_group = reference_group_short ,
#     compare_group = compare_group_short)
```

```

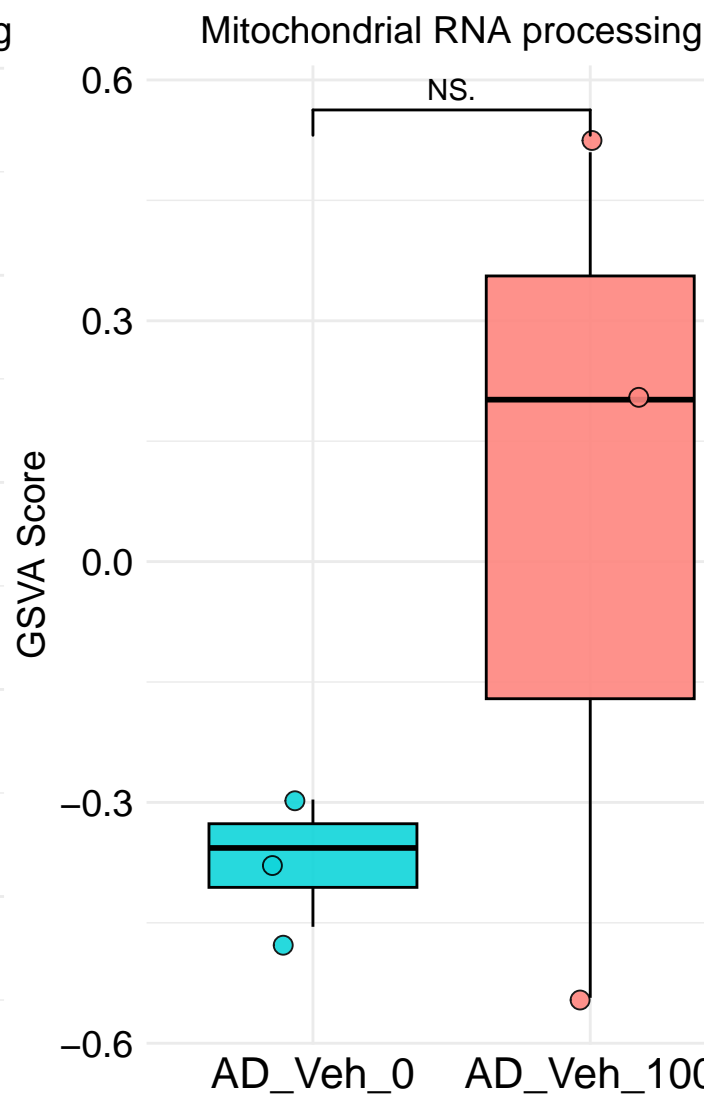
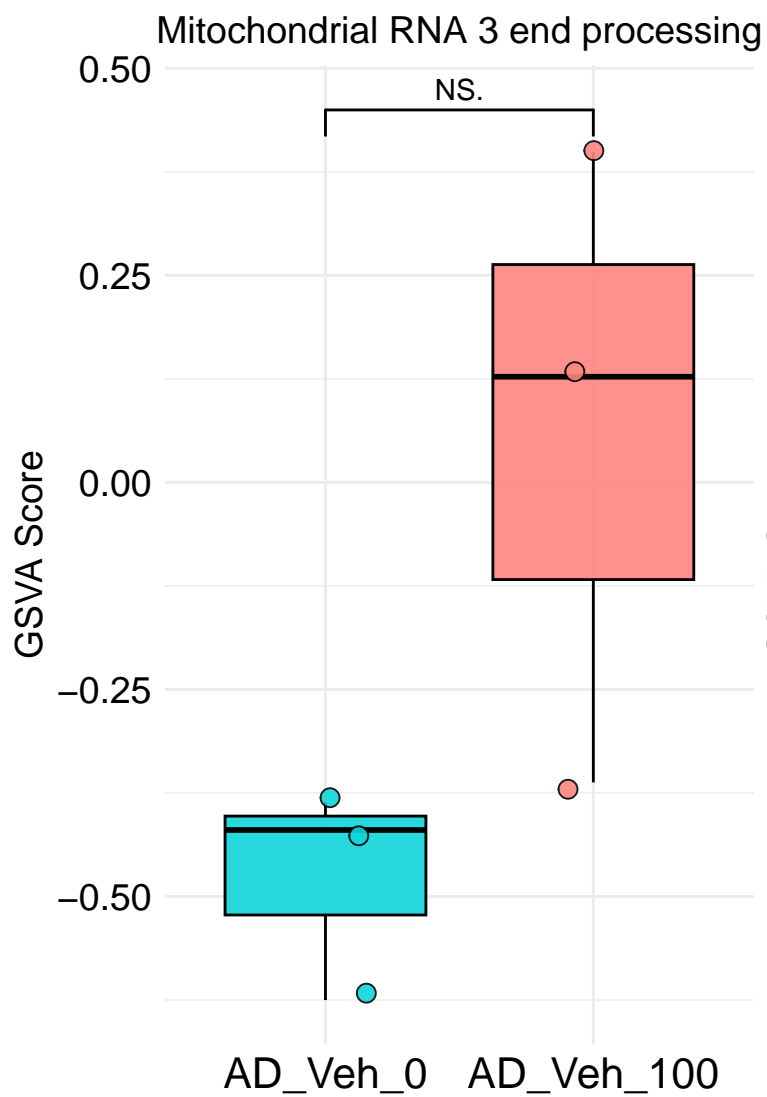
#
# }

box_plot_folder<- file.path(result_folder,"04-GSVA","Boxplot")
# create the folder
dir.create(box_plot_folder, showWarnings = FALSE)

# plot for the focus pathway
for (i in 1:2){
  pathway_name <- pathway_list$pathway[i]
  print(pathway_name)
  p<-plot_gsva_boxplot(gsva_matrix,
                       condition_list_label =condition_list_label,
                       pathway_name = pathway_name,
                       figure_folder = file.path(result_folder,"04-GSVA","Boxplot"),
                       file_name = paste0("GSVA_", pathway_name),
                       fig.height = 6, fig.width = 4,
                       reference_group =reference_group_short ,
                       compare_group = compare_group_short)
  print(p)
}

## [1] "GOBP_MITOCHONDRIAL_RNA_3_END_PROCESSING"
## [1] "GOBP_MITOCHONDRIAL_RNA_PROCESSING"

```

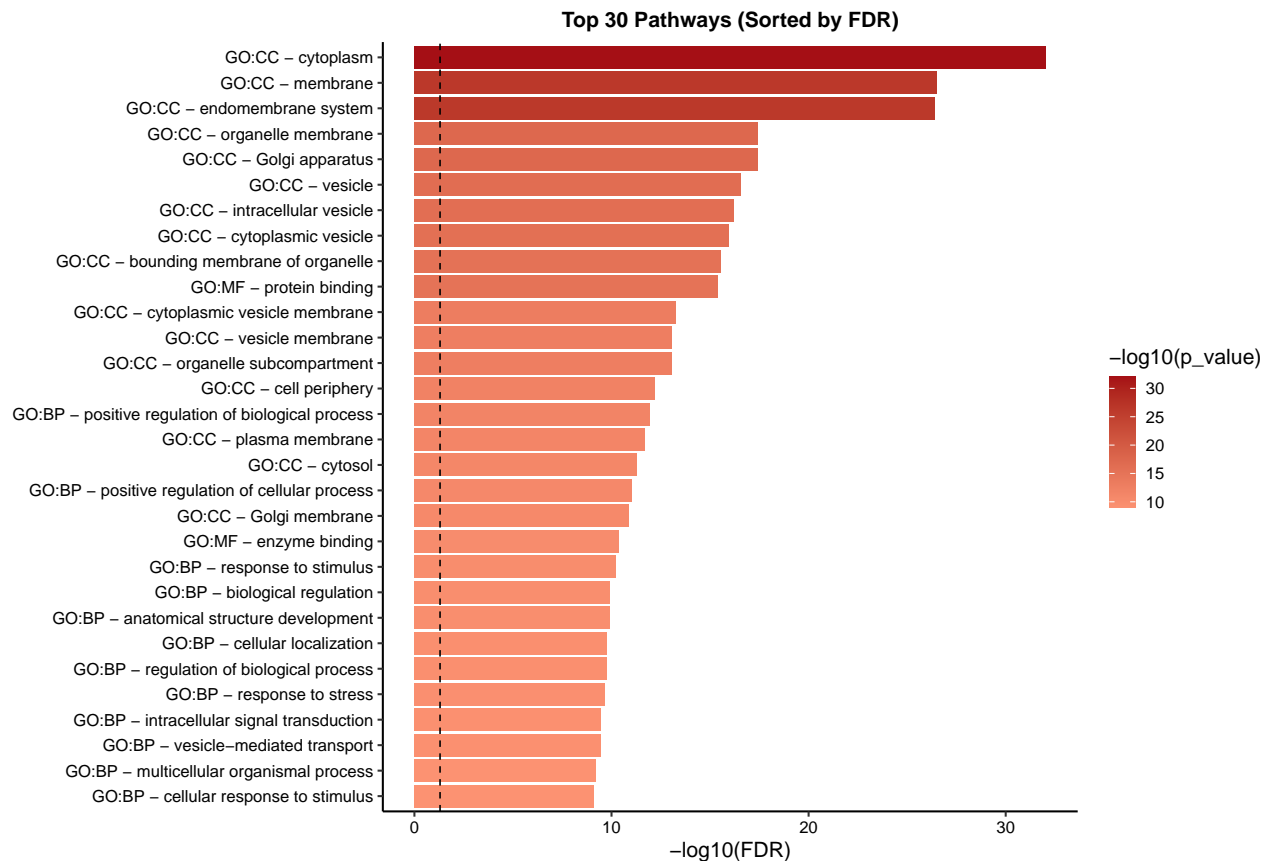
5. Pathway Enrichment Analysis

```
deg1 <- result_df %>% filter(padj < 0.05 & abs(log2FoldChange) > 1)
up_gene_1 <- deg1 %>% filter(log2FoldChange > 0) %>% pull(GeneName)
down_gene_1 <- deg1 %>% filter(log2FoldChange < 0) %>% pull(GeneName)
```

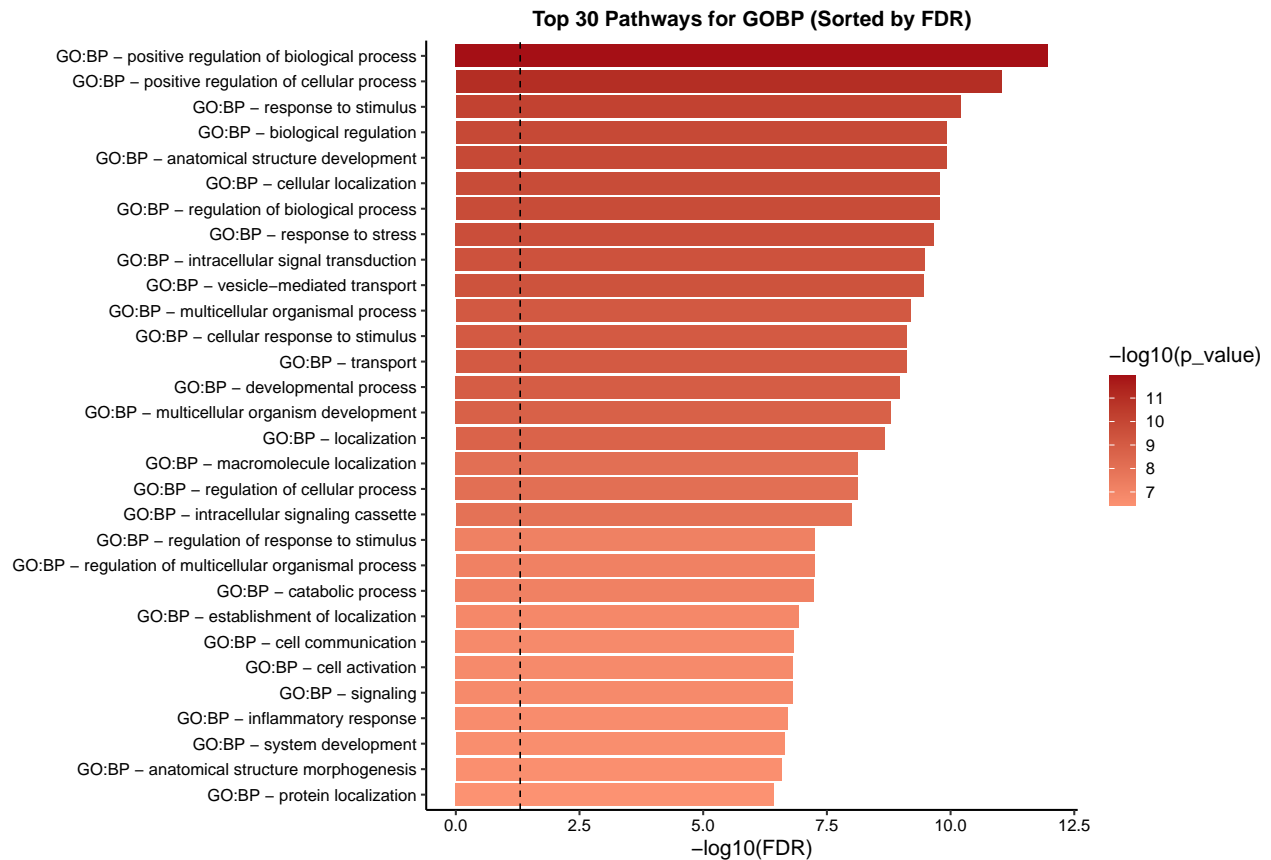
test for the function

```
Enrichment_analysis(gene_list = up_gene_1,
                    result_folder = file.path(result_folder, "03-Enrichment"),
                    file_name = "01-DEG_1.0_up", gene_name_mapping, flag = "Up")
```

```
## [1] "Enrichment analysis for 01-DEG_1.0_up "
```

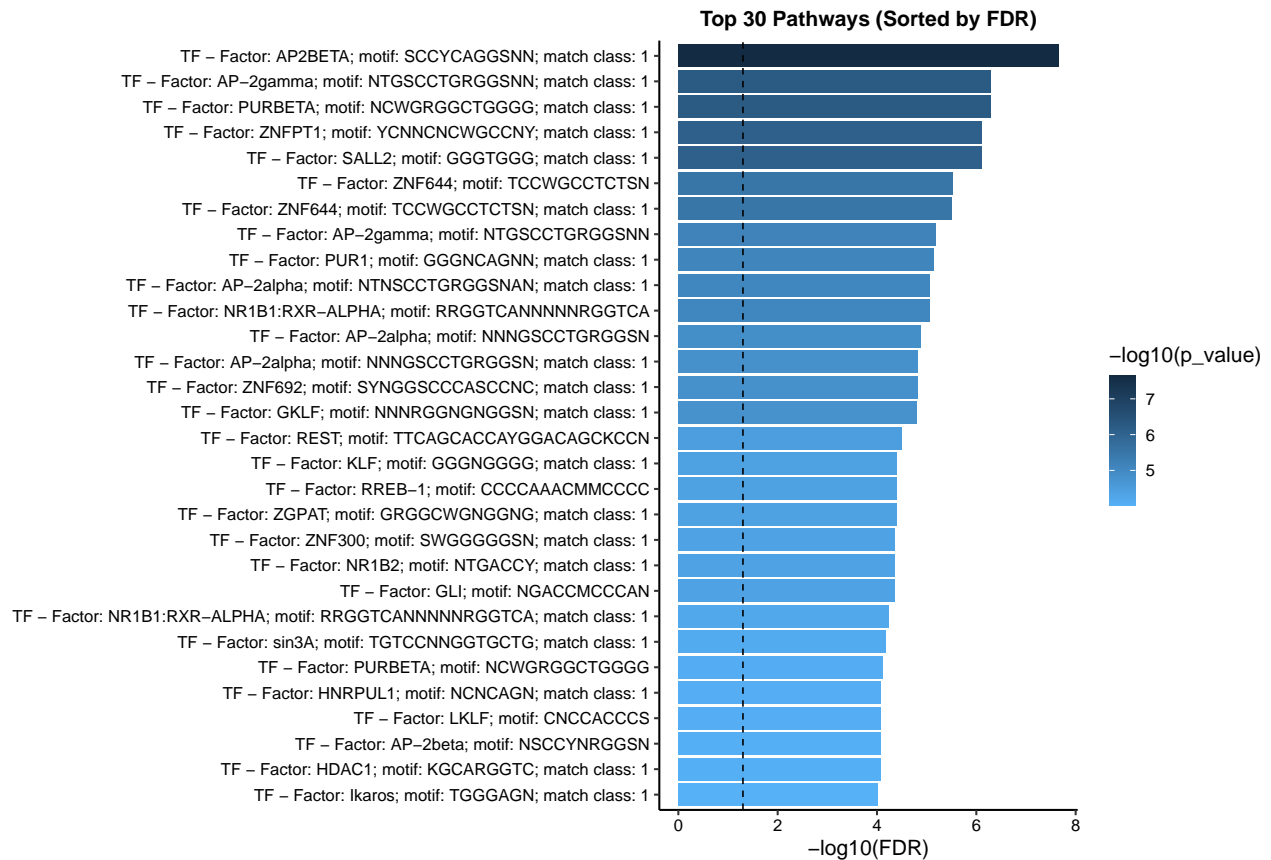


```
## [1] "Enrichment analysis for GOBP 01-DEG_1.0_up "
```



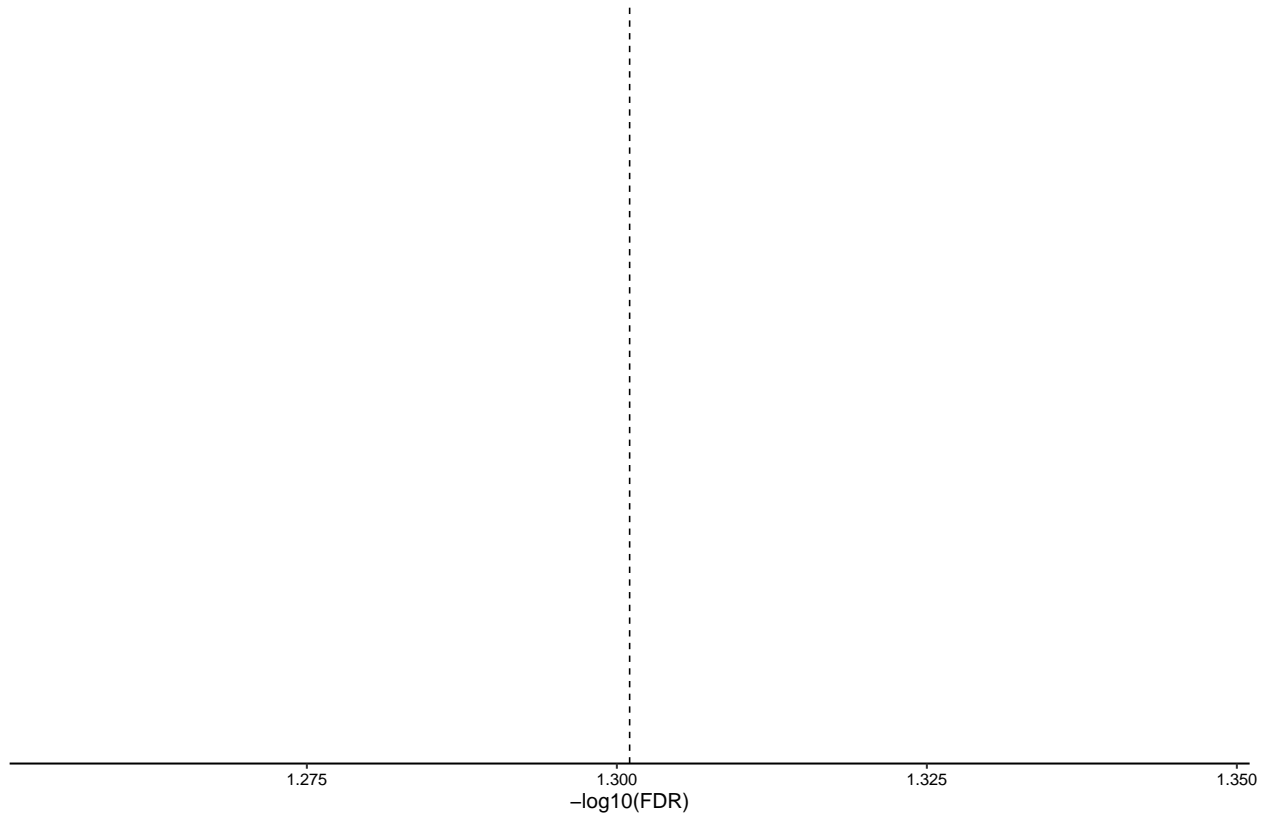
```
Enrichment_analysis(gene_list = down_gene_1,
                     result_folder = file.path(result_folder, "03-Enrichment"),
                     file_name = "01-DEG_1.0_down", gene_name_mapping, flag = "Down")
```

```
## [1] "Enrichment analysis for 01-DEG_1.0_down "
```



```
## [1] "Enrichment analysis for GOBP 01-DEG_1.0_down "
```

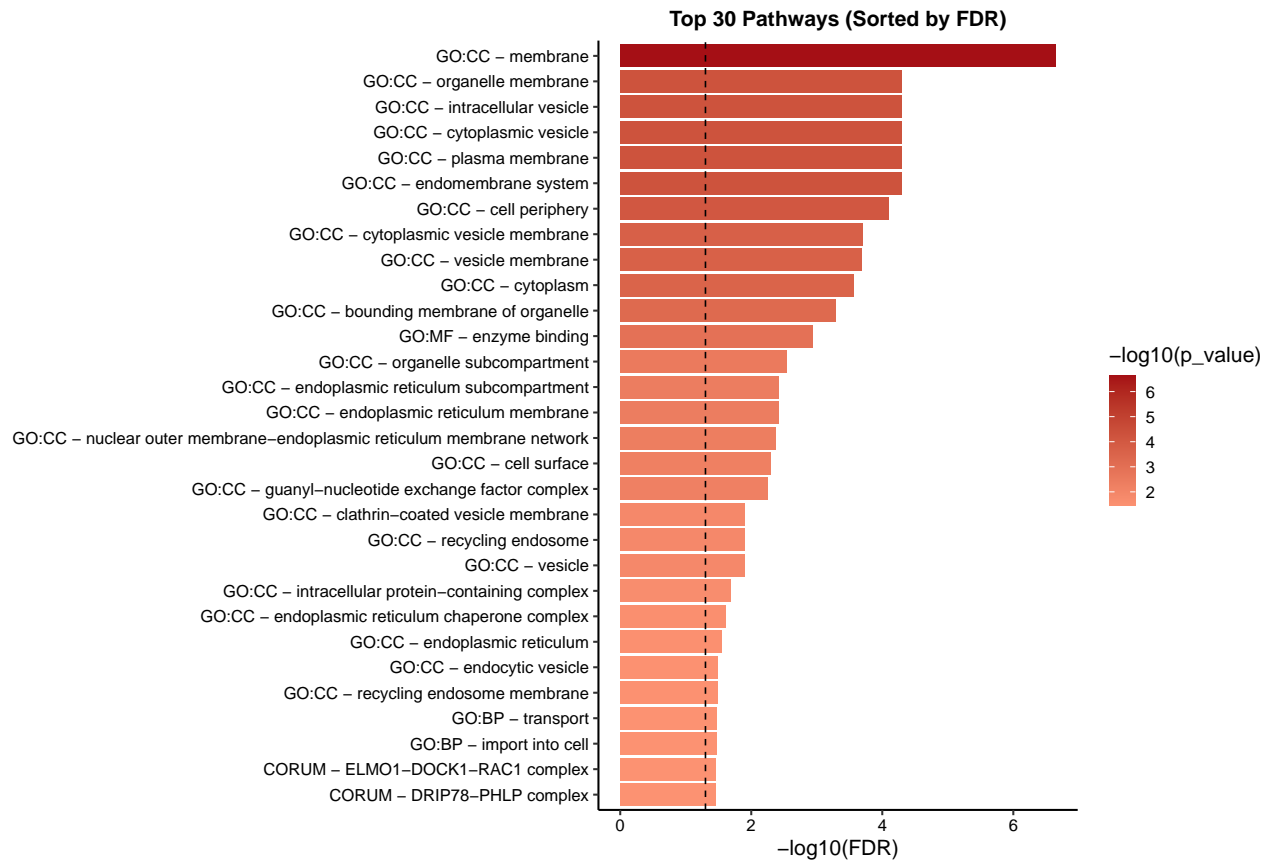
Top 30 Pathways for GOBP (Sorted by FDR)



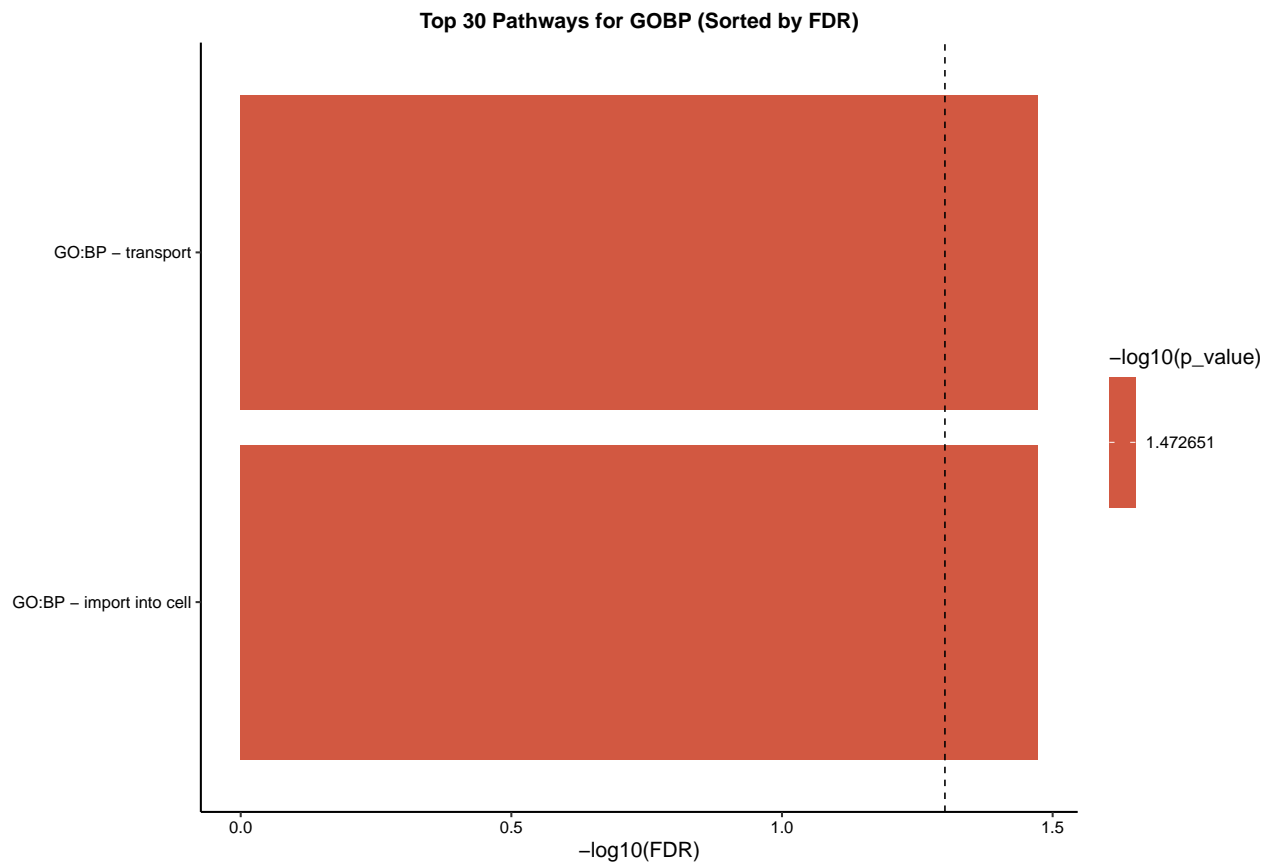
```
deg1.5 <- result_df %>% filter(padj < 0.05 & abs(log2FoldChange) > 1.5)
up_gene_1.5 <- deg1.5 %>% filter(log2FoldChange > 0) %>% pull(GeneName)
down_gene_1.5 <- deg1.5 %>% filter(log2FoldChange < 0) %>% pull(GeneName)

# test for the function
Enrichment_analysis(gene_list = up_gene_1.5,
                    result_folder = file.path(result_folder, "03-Enrichment"),
                    file_name = "02-DEG_1.5_up", gene_name_mapping, flag = "Up")

## [1] "Enrichment analysis for 02-DEG_1.5_up "
```

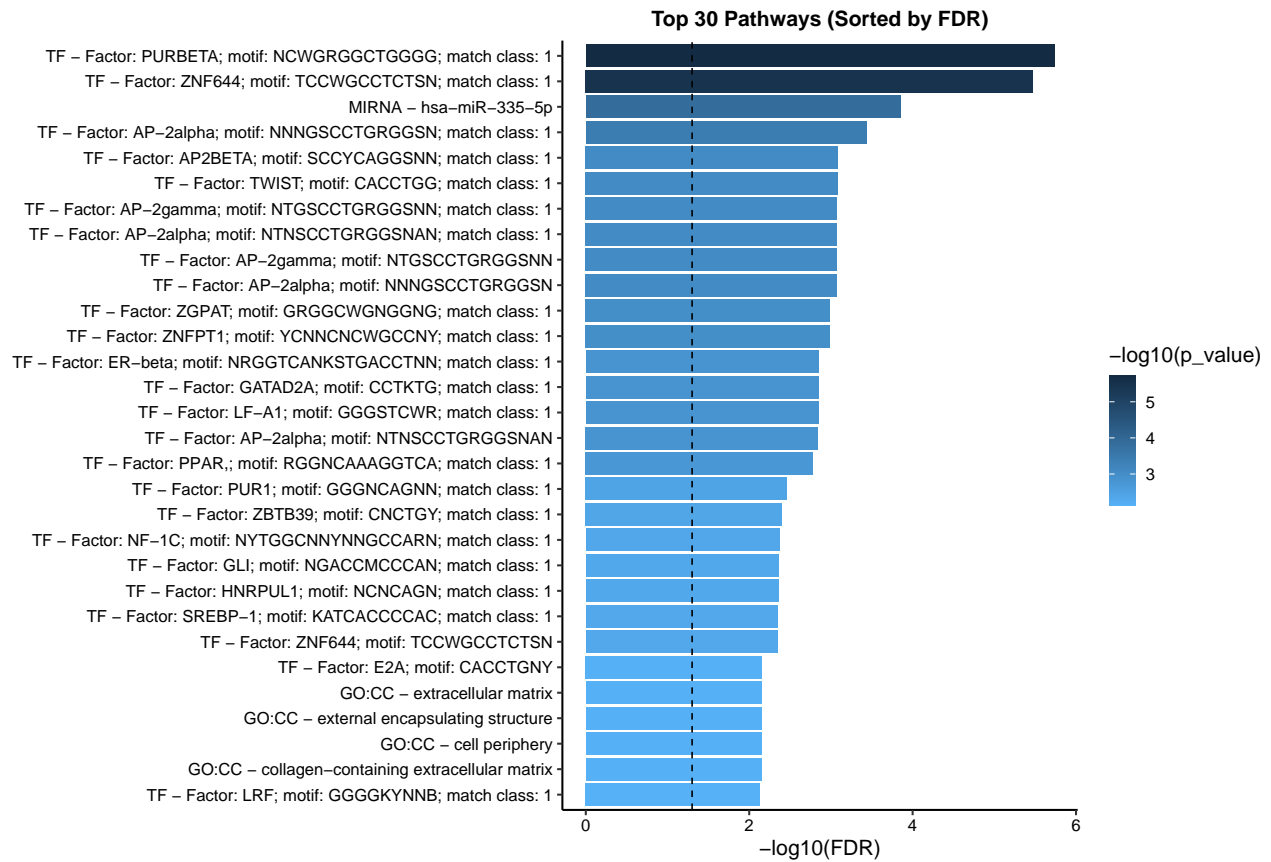


```
## [1] "Enrichment analysis for GOBP 02-DEG_1.5_up "
```



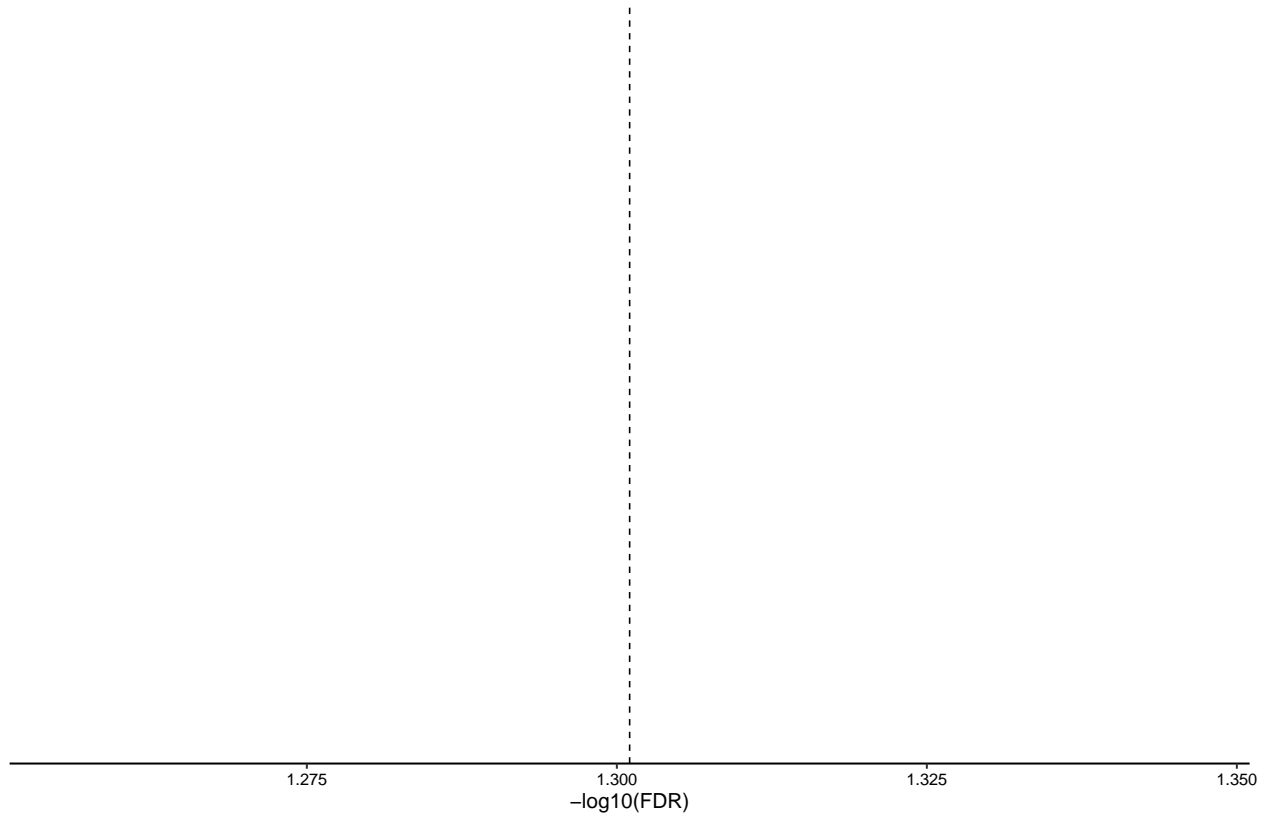
```
Enrichment_analysis(gene_list = down_gene_1.5,  
                    result_folder = file.path(result_folder, "03-Enrichment"),  
                    file_name = "02-DEG_1.5_down", gene_name_mapping, flag = "Down")
```

```
## [1] "Enrichment analysis for 02-DEG_1.5_down "
```



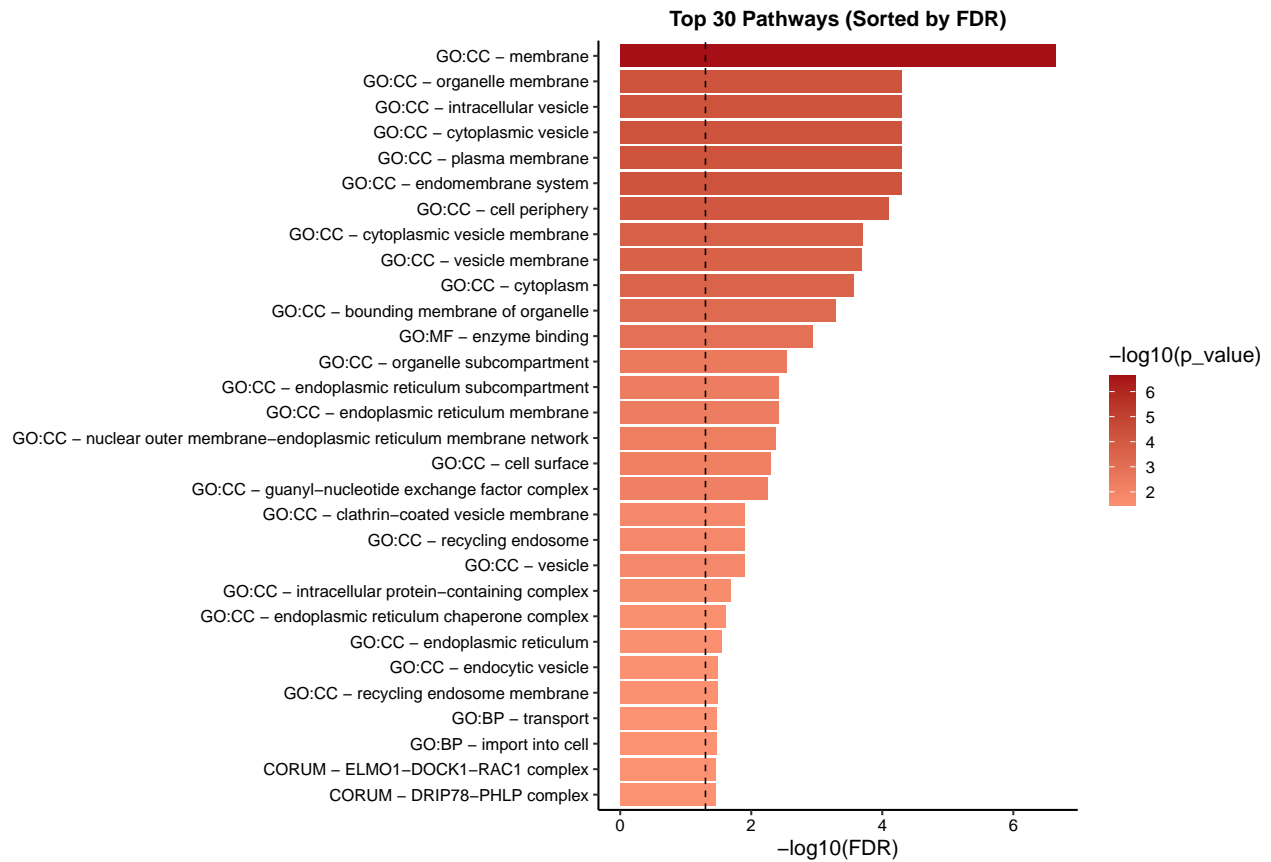
```
## [1] "Enrichment analysis for GOBP 02-DEG_1.5_down "
```


Top 30 Pathways for GOBP (Sorted by FDR)

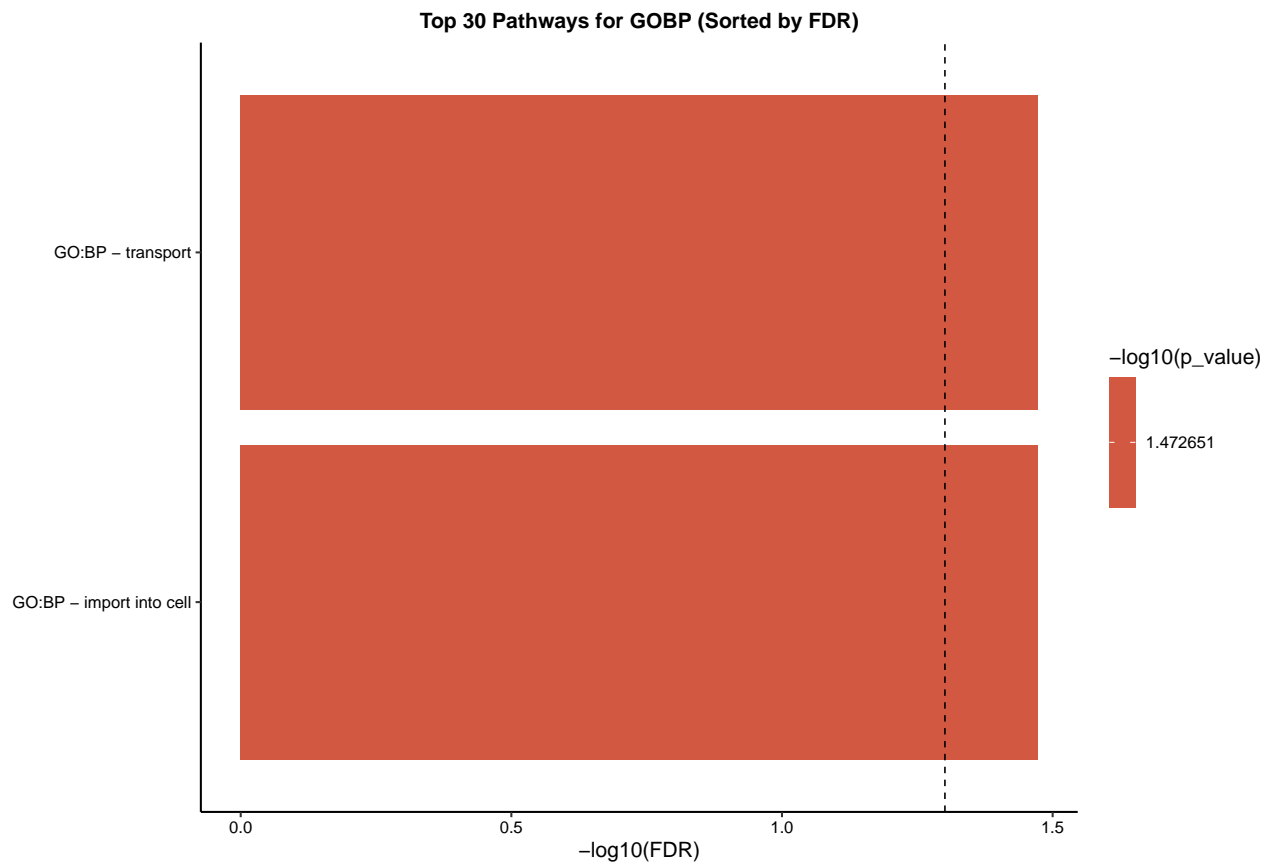


```
deg2 <- result_df %>% filter(padj < 0.05 & abs(log2FoldChange) > 1.5)
up_gene_2 <- deg2 %>% filter(log2FoldChange > 0) %>% pull(GeneName)
down_gene_2 <- deg2 %>% filter(log2FoldChange < 0) %>% pull(GeneName)
result_folder = result_folder_all
# test for the function
Enrichment_analysis(gene_list = up_gene_2,
  result_folder = file.path(result_folder, "03-Enrichment"),
  file_name = "03-DEG_2_up", gene_name_mapping, flag = "Up")

## [1] "Enrichment analysis for 03-DEG_2_up "
```

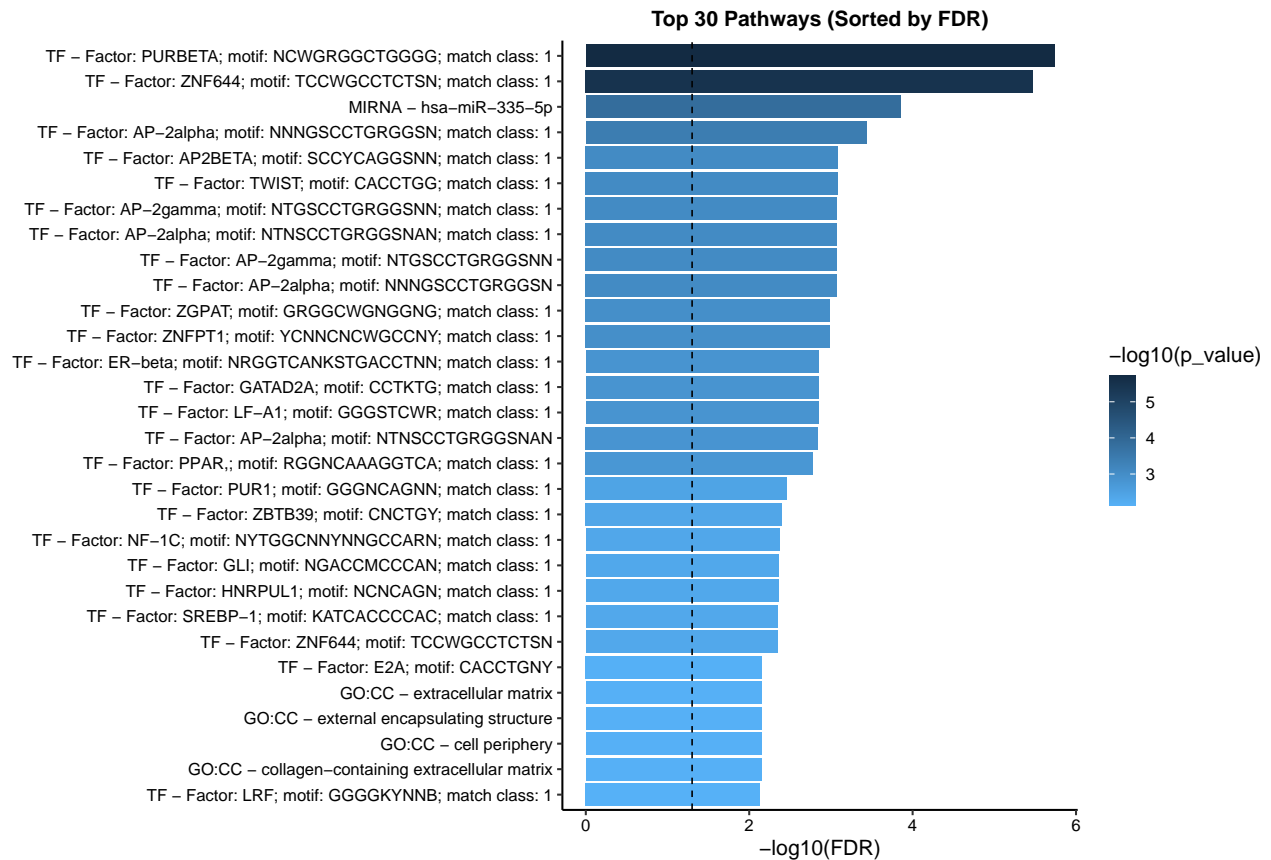


```
## [1] "Enrichment analysis for GOBP 03-DEG_2_up "
```



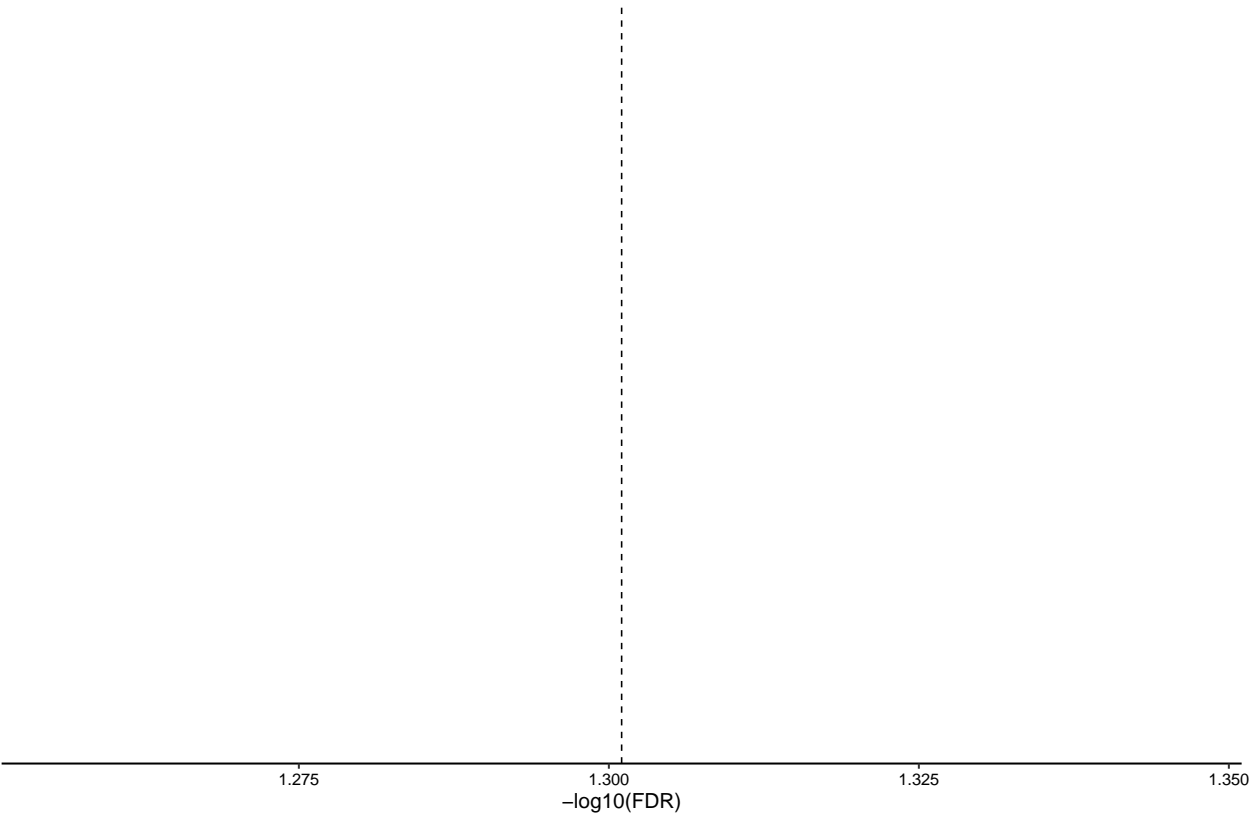
```
Enrichment_analysis(gene_list = down_gene_2,
                    result_folder = file.path(result_folder, "03-Enrichment"),
                    file_name = "03-DEG_2_down", gene_name_mapping, flag = "Down")
```

```
## [1] "Enrichment analysis for 03-DEG_2_down "
```



```
## [1] "Enrichment analysis for GOBP 03-DEG_2_down "
```

Top 30 Pathways for GOBP (Sorted by FDR)



Session information

```
sessionInfo()
```

```
## R version 4.4.0 (2024-04-24)
## Platform: aarch64-apple-darwin20
## Running under: macOS Sonoma 14.3.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib; LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] parallel stats4 stats graphics grDevices utils datasets
## [8] methods base
##
## other attached packages:
## [1] GSEABase_1.66.0 graph_1.82.0
## [3] annotate_1.82.0 XML_3.99-0.18
## [5] extrafont_0.19 ggsignif_0.6.4
## [7] patchwork_1.3.0 decoupleR_2.10.0
## [9] GSVA_1.52.3 BiocParallel_1.38.0
## [11] edgeR_4.2.2 limma_3.60.6
## [13] GenomicFeatures_1.56.0 biomaRt_2.60.1
## [15] gprofiler2_0.2.3 RColorBrewer_1.1-3
## [17] data.table_1.16.4 org.Hs.eg.db_3.19.1
## [19] AnnotationDbi_1.66.0 clusterProfiler_4.12.6
## [21] ggfortify_0.4.17 pheatmap_1.0.12
## [23] EnhancedVolcano_1.22.0 ggrepel_0.9.6
## [25] apeglm_1.26.1 DESeq2_1.44.0
## [27] SummarizedExperiment_1.34.0 Biobase_2.64.0
## [29] MatrixGenerics_1.16.0 matrixStats_1.5.0
## [31] reshape2_1.4.4 Matrix_1.7-2
## [33] Signac_1.14.0 Seurat_5.2.1
## [35] SeuratObject_5.0.2 sp_2.2-0
## [37] rtracklayer_1.64.0 GenomicRanges_1.56.2
## [39] GenomeInfoDb_1.40.1 IRanges_2.38.1
## [41] S4Vectors_0.42.1 BiocGenerics_0.50.0
## [43] knitr_1.49 lubridate_1.9.4
## [45] forcats_1.0.0 stringr_1.5.1
## [47] dplyr_1.1.4 purrr_1.0.4
## [49] readr_2.1.5 tidyr_1.3.1
## [51] tibble_3.2.1 ggplot2_3.5.1
## [53] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
## [1] SpatialExperiment_1.14.0 R.methodsS3_1.8.2
## [3] progress_1.2.3 goftest_1.2-3
```

##	[5]	HDF5Array_1.32.1	Biostrings_2.72.1
##	[7]	vctrs_0.6.5	spatstat.random_3.3-2
##	[9]	digest_0.6.37	png_0.1-8
##	[11]	deldir_2.0-4	parallelly_1.42.0
##	[13]	magick_2.8.5	MASS_7.3-64
##	[15]	httpuv_1.6.15	qvalue_2.36.0
##	[17]	withr_3.0.2	xfun_0.51
##	[19]	ggfun_0.1.8	survival_3.8-3
##	[21]	memoise_2.0.1	gson_0.1.0
##	[23]	systemfonts_1.2.1	ragg_1.3.3
##	[25]	tidytree_0.4.6	zoo_1.8-12
##	[27]	pbapply_1.7-2	R.oo_1.27.0
##	[29]	prettyunits_1.2.0	KEGGREST_1.44.1
##	[31]	promises_1.3.2	httr_1.4.7
##	[33]	restfulr_0.0.15	rhdf5filters_1.16.0
##	[35]	globals_0.16.3	fitdistrplus_1.2-2
##	[37]	rhdf5_2.48.0	rstudioapi_0.17.1
##	[39]	UCSC.utils_1.0.0	miniUI_0.1.1.1
##	[41]	generics_0.1.3	DOSE_3.30.5
##	[43]	curl_6.2.1	zlibbioc_1.50.0
##	[45]	ScaledMatrix_1.12.0	ggraph_2.2.1
##	[47]	polyclip_1.10-7	GenomeInfoDbData_1.2.12
##	[49]	SparseArray_1.4.8	xtable_1.8-4
##	[51]	evaluate_1.0.3	S4Arrays_1.4.1
##	[53]	BiocFileCache_2.12.0	hms_1.1.3
##	[55]	irlba_2.3.5.1	colorspace_2.1-1
##	[57]	filelock_1.0.3	ROCR_1.0-11
##	[59]	reticulate_1.40.0	spatstat.data_3.1-4
##	[61]	magrittr_2.0.3	lmtest_0.9-40
##	[63]	later_1.4.1	viridis_0.6.5
##	[65]	ggtree_3.12.0	lattice_0.22-6
##	[67]	spatstat.geom_3.3-5	future.apply_1.11.3
##	[69]	scattermore_1.2	shadowtext_0.1.4
##	[71]	cowplot_1.1.3	RcppAnnoy_0.0.22
##	[73]	pillar_1.10.1	nlme_3.1-167
##	[75]	compiler_4.4.0	beachmat_2.20.0
##	[77]	RSpectra_0.16-2	stringi_1.8.4
##	[79]	tensor_1.5	GenomicAlignments_1.40.0
##	[81]	plyr_1.8.9	crayon_1.5.3
##	[83]	abind_1.4-8	BiocIO_1.14.0
##	[85]	gridGraphics_0.5-1	emdbbook_1.3.13
##	[87]	locfit_1.5-9.11	graphlayouts_1.2.2
##	[89]	bit_4.5.0.1	fastmatch_1.1-6
##	[91]	textshaping_1.0.0	codetools_0.2-20
##	[93]	BiocSingular_1.20.0	plotly_4.10.4
##	[95]	mime_0.12	splines_4.4.0
##	[97]	Rcpp_1.0.14	fastDummies_1.7.5
##	[99]	sparseMatrixStats_1.16.0	dbplyr_2.5.0
##	[101]	Rttf2pt1_1.3.12	blob_1.2.4
##	[103]	here_1.0.1	fs_1.6.5
##	[105]	listenv_0.9.1	ggplotify_0.1.2
##	[107]	statmod_1.5.0	tzdb_0.4.0
##	[109]	tweenr_2.0.3	pkgconfig_2.0.3
##	[111]	tools_4.4.0	cachem_1.1.0

## [113] RSQLite_2.3.9	viridisLite_0.4.2
## [115] DBI_1.2.3	numDeriv_2016.8-1.1
## [117] fastmap_1.2.0	rmarkdown_2.29
## [119] scales_1.3.0	grid_4.4.0
## [121] ica_1.0-3	Rsamtools_2.20.0
## [123] coda_0.19-4.1	dotCall64_1.2
## [125] RANN_2.6.2	farver_2.1.2
## [127] tidygraph_1.3.1	scatterpie_0.2.4
## [129] yaml_2.3.10	cli_3.6.4
## [131] lifecycle_1.0.4	uwot_0.2.2
## [133] mvtnorm_1.3-3	timechange_0.3.0
## [135] gtable_0.3.6	rjson_0.2.23
## [137] ggthemes_0.5.6	progressr_0.15.1
## [139] ape_5.8-1	jsonlite_1.9.0
## [141] RcppHNSW_0.6.0	bitops_1.0-9
## [143] bit64_4.6.0-1	Rtsne_0.17
## [145] yulab.utils_0.2.0	spatstat.utils_3.1-2
## [147] bdsmatrix_1.3-7	GOSemSim_2.30.2
## [149] spatstat.univar_3.1-1	R.utils_2.12.3
## [151] lazyeval_0.2.2	shiny_1.10.0
## [153] htmltools_0.5.8.1	enrichplot_1.24.4
## [155] GO.db_3.19.1	sctransform_0.4.1
## [157] rappdirs_0.3.3	tinytex_0.55
## [159] glue_1.8.0	spam_2.11-1
## [161] httr2_1.1.0	XVector_0.44.0
## [163] RCurl_1.98-1.16	rprojroot_2.0.4
## [165] treeio_1.28.0	gridExtra_2.3
## [167] extrafontdb_1.0	igraph_2.1.4
## [169] R6_2.6.1	SingleCellExperiment_1.26.0
## [171] labeling_0.4.3	RcppRoll_0.3.1
## [173] cluster_2.1.8	bbmle_1.0.25.1
## [175] Rhdf5lib_1.26.0	aplot_0.2.4
## [177] DelayedArray_0.30.1	tidyselect_1.2.1
## [179] ggforce_0.4.2	xml2_1.3.6
## [181] future_1.34.0	rsvd_1.0.5
## [183] munsell_0.5.1	KernSmooth_2.23-26
## [185] htmlwidgets_1.6.4	fgsea_1.30.0
## [187] rlang_1.1.5	spatstat.sparse_3.1-0
## [189] spatstat.explore_3.3-4	