Gene Ontology Analysis and Heatmap Visualization

Load Data

```
# Read the result table
genes <- read.table('/Users/ying-chulo/Desktop/Rmarkdown/genes.txt')

result_table <- read.table('/Users/ying-chulo/Desktop/Rmarkdown/96hrs_Col_vs_96hrs_mon.edgeR.DE_results

# preview the table
head(result_table)
```

```
##             sampleA   sampleB     logFC   logCPM       PValue          FDR
## AT1G52400 96hrs_Col 96hrs_mon  7.904215 10.83800 2.212207e-08 2.566160e-06
## AT1G20620 96hrs_Col 96hrs_mon  4.860405 10.77173 1.750067e-07 1.015039e-05
## AT2G39730 96hrs_Col 96hrs_mon -2.216494 12.35439 8.347374e-06 3.227651e-04
## AT3G02470 96hrs_Col 96hrs_mon  3.550746 10.70089 4.998708e-05 1.449625e-03
## AT3G44310 96hrs_Col 96hrs_mon  2.859266 10.92458 5.564624e-04 1.290993e-02
## AT1G29930 96hrs_Col 96hrs_mon -3.675529 11.10411 7.946178e-04 1.536261e-02
```

Process the data and filter by logFC

```
# Generate results table
TAIR <- rownames(result_table)
res <- cbind(result_table, TAIR)

# Read GO annotation file (assuming it's already loaded as `genes`)
go_data <- genes

# Merge go_data and res based on TAIR
merged_data <- merge(x = res, y = go_data, by = "TAIR")

# Filter for top genes
merged_res_at_top <- merged_data[abs(merged_data$logFC) > 2,]
merged_res_at_top <- merged_res_at_top[order(merged_res_at_top$logFC, decreasing = TRUE),]

# check merged_res_at_top
head(merged_res_at_top)
```

```
##         TAIR   sampleA   sampleB    logFC    logCPM       PValue          FDR
## 28 AT1G52400 96hrs_Col 96hrs_mon 7.904215 10.838000 2.212207e-08 2.566160e-06
## 1  AT1G02930 96hrs_Col 96hrs_mon 5.446228  9.776305 6.563243e-02 2.944645e-01
## 31 AT1G61520 96hrs_Col 96hrs_mon 5.132549 10.237654 1.287405e-01 4.031616e-01
## 15 AT1G20620 96hrs_Col 96hrs_mon 4.860405 10.771727 1.750067e-07 1.015039e-05
## 84 AT4G13340 96hrs_Col 96hrs_mon 4.173100  9.437223 5.024876e-01 8.095633e-01
## 53 AT3G02470 96hrs_Col 96hrs_mon 3.550746 10.700891 4.998708e-05 1.449625e-03
##            GO EVIDENCE ONTOLOGY
## 28 GO:0000325      HDA       CC
```

```
## 1  GO:0000325      HDA       CC
## 31 GO:0003729      IDA       MF
## 15 GO:0000325      HDA       CC
## 84 GO:0005199      ISS       MF
## 53 GO:0004014      IBA       MF
```

DAVID results and Annotation

```
# Read DAVID results
david <- read.delim("/Users/ying-chulo/Desktop/Rmarkdown/david.txt", sep="\t", header = T)
david <- subset(david, !is.na(Genes))
david <- subset(david, !is.na(Term))
colnames(david) <- c("Category", "Term", "Count", "X", "PValue", "Genes", "ListTotal", "PopHits", "PopT

# Initialize annGSEA data frame
annGSEA <- data.frame(row.names = merged_res_at_top$TAIR)

# Loop through each row in annGSEA
# The loop will generate a table with gene X GO terms, using 0 and 1 to indicate whether a gene exists

for (j in 1:length(rownames(annGSEA))) {
  gene <- rownames(annGSEA)[j]  # Extract the specific gene for this iteration
  # Loop through each row in david
  for (k in 1:nrow(david)) {
    if (gene %in% unlist(strsplit(david$Genes[k], ", "))) {
      # If gene is found in david$Genes, set corresponding cell in annGSEA to 1
      annGSEA[j, k] <- 1
    } else {
      # If gene is not found in david$Genes, set corresponding cell in annGSEA to 0
      annGSEA[j, k] <- 0
    }
  }
}

# Set column names of annGSEA
colnames(annGSEA) <- david$Term
```

```
# Remove genes with no overlapping terms
annGSEA <- annGSEA[, apply(annGSEA, 2, mean) != 0]

# Remove genes with 20 or more occurrences
annGSEA <- annGSEA[data.frame(rowSums(annGSEA)) >= 20,]

# Subset annGSEA based on matching Terms
annGSEA_subset <- annGSEA[, colnames(annGSEA) %in% david$Term]

# Remove the GO with 0 gene
annGSEA_subset_filtered <- annGSEA_subset[, colSums(annGSEA_subset != 0) > 0]

# check annGSEA
head(annGSEA_subset_filtered[, 1:2], n = 3)
```

```
##           GO:0005829~cytosol GO:0005515~protein binding
## AT1G52400                  0                          0
```

```
## AT1G02930                  1                          0
## AT1G61520                  0                          1
```

Prepare Heatmap Annotation

```
# Create the color bar for -log10(Benjamini enrichment Q value)
benjamini_values <- -log10(david[which(colnames(annGSEA_subset_filtered) %in% david$Term), 'Benjamini']]
benjamini_values[is.infinite(benjamini_values)] <- NA # find NA otherwise the haTerms will return error
dfMinusLog10BenjaminiTerms <- data.frame(benjamini_values)
colnames(dfMinusLog10BenjaminiTerms) <- 'Enrichment\nterm score'

# Create Heatmap Annotation
haTerms <- HeatmapAnnotation(
  df = dfMinusLog10BenjaminiTerms,
  annotation_name_align = FALSE,
  Term = anno_text(
    colnames(annGSEA_subset_filtered),
    rot = 90,
    gp = gpar(fontsize = 10)
  )
)
```

Prepare Gene Annotation

Plot Heatmap

```
# Plot
hmapGSEA <- Heatmap(annGSEA_subset_filtered,
                    name = 'DAVID GO enrichment',
                    split = dfGeneAnno[,2],
                    col = c('0' = 'white', '1' = 'forestgreen'),
                    rect_gp = gpar(col = 'grey85'),
                    cluster_rows = TRUE,
                    show_row_dend = TRUE,
                    row_title = 'Top Genes',
                    row_title_side = 'left',
                    row_title_gp = gpar(fontsize = 10, fontface = 'bold'),
                    row_title_rot = 90,
                    show_row_names = TRUE,
                    row_names_gp = gpar(fontsize = 9, fontface = 'bold'),
                    row_names_side = 'left',
                    row_dend_width = unit(35, 'mm'),
                    cluster_columns = TRUE,
                    show_column_dend = TRUE,
                    column_title = 'Enriched terms',
                    column_title_side = 'top',
                    column_title_gp = gpar(fontsize = 10, fontface = 'bold'),
                    column_title_rot = 0,
                    show_column_names = FALSE,
                    show_heatmap_legend = FALSE,
                    clustering_distance_columns = 'euclidean',
                    clustering_method_columns = 'ward.D2',
                    clustering_distance_rows = 'euclidean',
```

```r
                    clustering_method_rows = 'ward.D2',
                    bottom_annotation = haTerms)
```

```
## Warning: The input is a data frame-like object, convert it to a matrix.
```

```r
draw(hmapGSEA + haGenes,
     heatmap_legend_side = 'right',
     annotation_legend_side = 'right')
```