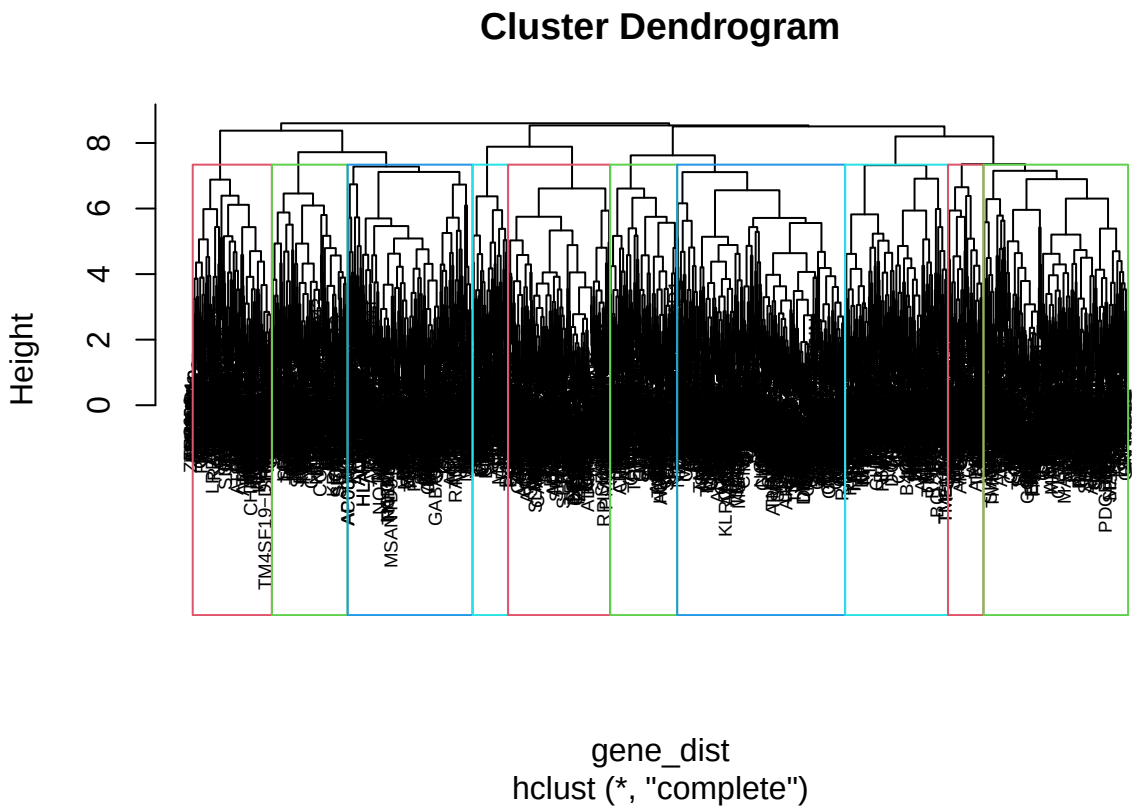
h_clustering

```
source_rmd = function(file, ...) {
  tmp_file = tempfile(fileext=".R")
  on.exit(unlink(tmp_file), add = TRUE)
  knitr::purl(file, output=tmp_file)
  source(file = tmp_file, ...)
}

options(knitr.duplicate.label = "allow")
source_rmd("rawdata_normalization.rmd")
source_rmd("functions.rmd")
```

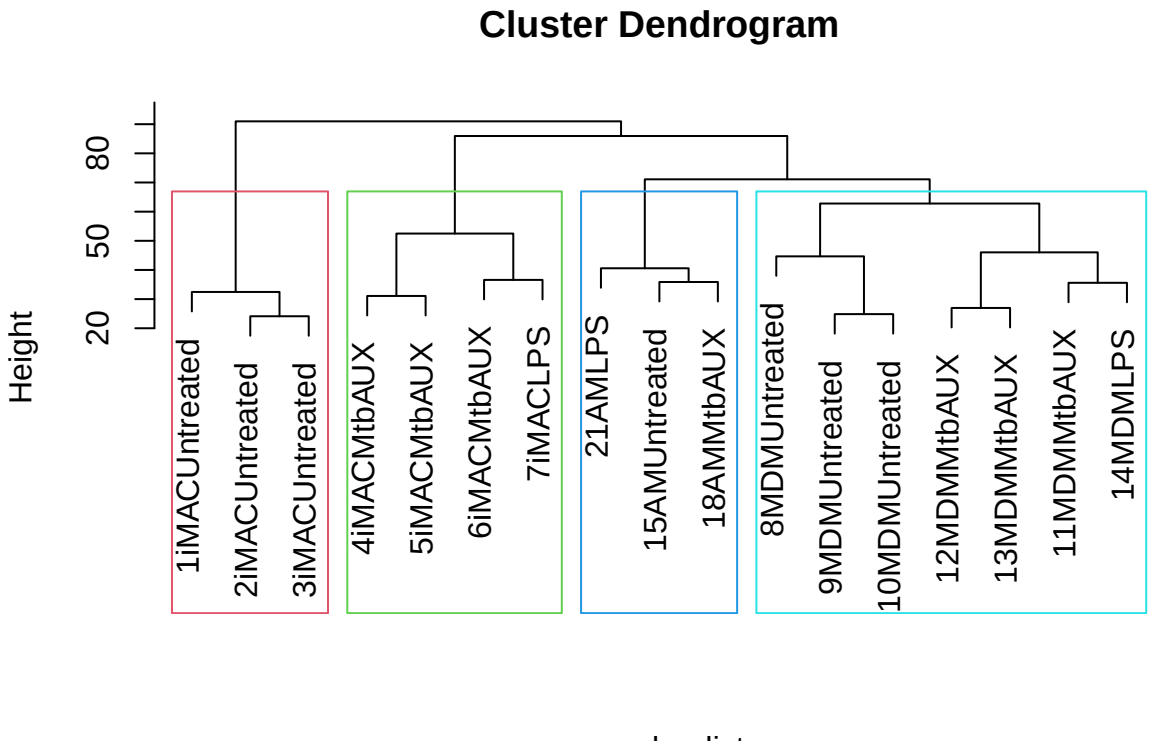
```
# Filter the results dataframes for SDEGS
sdeg_am <- sdeg_extraction(results$AM.MtbAUXvsAM.Untreated)
sdeg_mdm <- sdeg_extraction(results$MDM.MtbAUXvsMDM.Untreated)
sdeg_imac <- sdeg_extraction(results$iMACs.MtbAUXvsiMACs.Untreated)
sdeg_thp1 <- sdeg_extraction(results$THP1.MtbAUXvsTHP1.Untreated)

# Create a shared DF of all SDEGs
all_sdegs <- bind_rows(sdeg_imac, sdeg_am, sdeg_mdm) %>% dplyr::select(-2)


all_sdegs2 <- bind_rows(sdeg_imac, sdeg_am, sdeg_mdm) %>% rownames_to_column("Gene_ID")

all_sdegs2$Gene_ID <- gsub("\\..*", "", all_sdegs2$Gene_ID)
```

```
# z transformed should be euclidian
hclust_matrix <- inner_join(as.data.frame(z_transformed_norm_exp_am_rm)[1:17] %>% rownames_to_column("G
  distinct(gene_source, .keep_all = TRUE) %>%
  dplyr::select(-1) %>%
  filter(gene_source %in% unique(all_sdegs[1][[1]])) %>%
  column_to_rownames("gene_source") %>%
  as.matrix()


hclust_matrix2 <- hclust_matrix %>% t()

gene_dist <- dist(hclust_matrix, method = "euclidean") #euclidian


gene_hclust <- hclust(gene_dist, method = "complete")


sample_dist <- dist(hclust_matrix2)
sample_hclust <- hclust(sample_dist, method = "complete")
```

```r
gene_cluster <- cutree(gene_hclust, k = 7)

# Make dataframe of the genes and their respective clusters for enrichment analysis
gene_cluster_df <- gene_cluster %>% enframe()
names(gene_cluster_df) <- c("gene_source", "cluster")

plot(gene_hclust, cex = 0.6)
rect.hclust(gene_hclust, k = 10, border = 2:5)
```



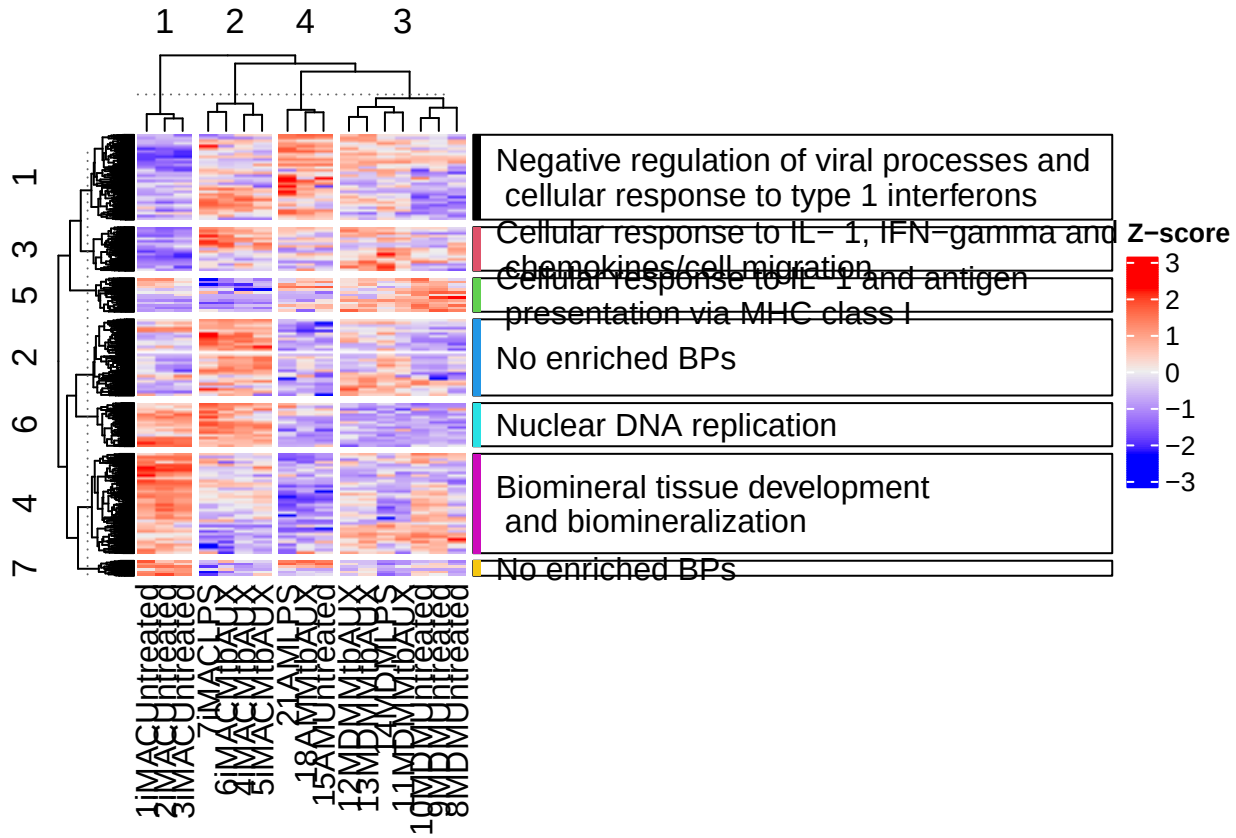

```r
sample_cluster <- cutree(sample_hclust, k = 4)

plot(sample_hclust, cex = 1)
rect.hclust(sample_hclust, k = 4, border = 2:5)
```




```
cluster_go_enrichment <- function(deg_df) {
  # all genes for background
  all_background <- deg_df$Gene_ID %>% as.character()

  ego_list <- list()

  for (cluster_nr in 1:max(gene_cluster_df$cluster)) {
    sig_genes <-
      all_sdegs2 %>% filter(gene_source %in% filter(gene_cluster_df, cluster == cluster_nr)[1][[1]])
    sig_genes <- sig_genes$Gene_ID


    ego <- enrichGO(
      gene = sig_genes,
      universe = all_background,
      keyType = "ENSEMBL",
      OrgDb = org.Hs.eg.db,
      ont = "BP",
      maxGSSize = 100,
      pAdjustMethod = "BH",
      qvalueCutoff = 0.05,
      readable = TRUE
    )

    ego_list[cluster_nr] <- ego
```

```
  }
  ego_list
}

ego_results <- cluster_go_enrichment(results$iMACs.MtbAUXvsiMACs.Untreated)

saveRDS(ego_results, "h_clustering_ego.RData")

dotplot(ego_results[[1]])
dotplot(ego_results[[2]])
dotplot(ego_results[[3]])
dotplot(ego_results[[4]])
#dotplot(ego_results[[5]])
dotplot(ego_results[[6]])
#dotplot(ego_results[[7]])
#dotplot(ego_results[[8]])
#dotplot(ego_results[[9]])
```

```
#Heatmap(hclust_matrix, show_row_names = FALSE, split = gene_cluster, column_split = factor(sample_clus

text_list = list(
  text2 = "Negative regulation of viral processes and \n cellular response to type 1 interferons",
  text3 = "Cellular response to IL- 1, IFN-gamma and \n chemokines/cell migration",
  text1 = "Cellular response to IL-1 and antigen \n presentation via MHC class I",
  text6 = "No enriched BPs",
  text4 = "Nuclear DNA replication",
  text5 = "Biomineral tissue development \n and biomineralization ",
  text7 = "No enriched BPs"

)

ha = rowAnnotation(Go = anno_empty(border = TRUE,
    width = max_text_width(unlist(text_list)) + unit(1, "mm")))

Heatmap(
  hclust_matrix,
  show_row_names = FALSE,
  split = factor(gene_cluster),
  column_split = factor(sample_cluster, levels = c(1, 3, 2, 4)),
  name = "Z-score",
  right_annotation = ha
)

for(i in 1:7) {
    decorate_annotation("Go", slice = i, {
        grid.rect(x = 0, width = unit(1, "mm"), gp = gpar(fill = i, col = NA), just = "left")
        grid.text(paste(text_list[[i]], collapse = "\n"), x = unit(3, "mm"), just = "left")
    })
}
```




```
extract_genes_in_term <- function(enrichment_df) {
  no_terms <- 10

  enriched_pathway_genes <- data.frame("term" = NULL, "genes" = NULL)

  for (term in 1:no_terms) {
    str <- str_split(enrichment_df$geneID[term], "/")
    enriched_pathway_genes <-
      rbind(
        data.frame("term" = enrichment_df$Description[term], "genes" = str[[1]]),
        enriched_pathway_genes
      )

  }
  names(enriched_pathway_genes) <- c("terms", "gene_source")
  enriched_pathway_genes
}
extract_genes_in_term(ego_results[[1]])
```

```
cluster_df_heatmap <- function(cluster_nr){
  cluster_df <- gene_cluster_df %>% filter(cluster == cluster_nr)
  cluster_df$cluster <- cluster_df$gene_source
  names(cluster_df) <- c("gene_source","original_names" )
  cluster_df
}
```

```r
cluster_1 <- cluster_df_heatmap(1)
cluster_2 <- cluster_df_heatmap(2)
cluster_3 <- cluster_df_heatmap(3)
cluster_4 <- cluster_df_heatmap(4)
cluster_6 <- cluster_df_heatmap(6)
cluster_7 <- cluster_df_heatmap(7)
cluster_8 <- cluster_df_heatmap(8)
```