Enrichment analysis

Function for redering rmd

```r
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, ...)
}
```

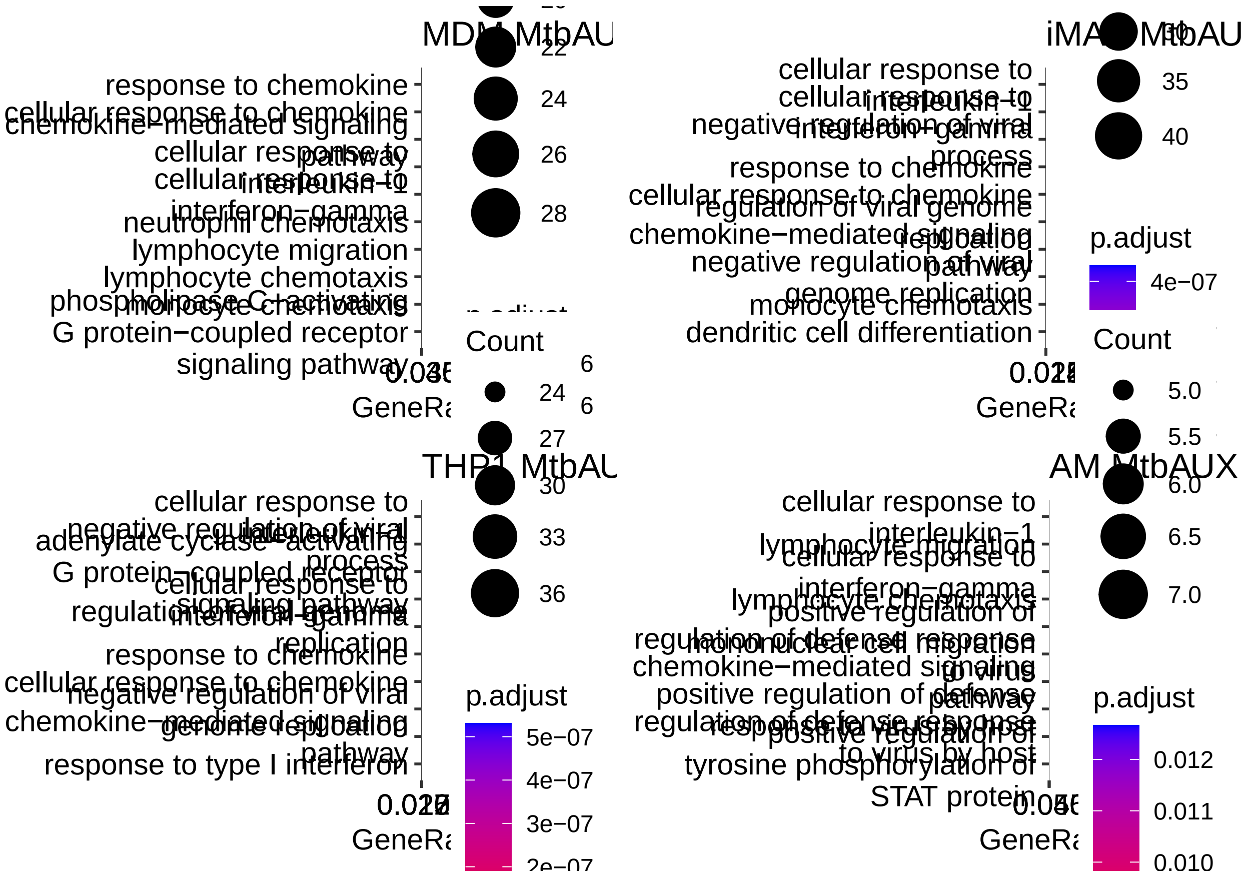
Reading in the raw data and the functions

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

```r
go_enrichment <- function(deg_df, adj = FALSE) {
  # all genes for background
  all_background <- deg_df$Gene_ID %>% as.character()

  #significantly DEGs
  if (adj == TRUE) {
    sig_genes <-
      deg_df %>% sdeg_extraction() %>% rownames_to_column(var = "Gene_ID")
    sig_genes <- sig_genes$Gene_ID

  } else{
    sig_genes <-
      deg_df %>% sdeg_extraction() %>% rownames_to_column(var = "Gene_ID")
    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
  )
  ## Output results from GO analysis to a table
  cluster_summary <- data.frame(ego)


  ego
}
```

```
mdm <- go_enrichment(results$MDM.MtbAUXvsMDM.Untreated, adj = FALSE)

iMAC <-
  go_enrichment(results$iMACs.MtbAUXvsiMACs.Untreated, adj = FALSE)

THP1 <-
  go_enrichment(results$THP1.MtbAUXvsTHP1.Untreated, adj = FALSE)

AM <- go_enrichment(results$AM.MtbAUXvsAM.Untreated, adj = FALSE)


# dotplot(AM, showCategory =10)
# dotplot(iMAC,showCategory =10)
# dotplot(mdm, showCategory =10)
# dotplot(THP1, showCategory =10)
```

```
width <- 30
font <- 11
one <-
  dotplot(mdm,
          showCategory = 10,
          title = "MDM MtbAUX vs Untreated",
          font.size = font) + scale_y_discrete(
            labels = function(egoBP)
              str_wrap(egoBP, width = width)
          )
two <-
  dotplot(iMAC,
          showCategory = 10,
          title = "iMAC MtbAUX vs Untreated",
          font.size = font) + scale_y_discrete(
            labels = function(egoBP)
              str_wrap(egoBP, width = width)
          )
three <-
  dotplot(THP1,
          showCategory = 10,
          title = "THP1 MtbAUX vs Untreated",
          font.size = font) + scale_y_discrete(
            labels = function(egoBP)
              str_wrap(egoBP, width = width)
          )
four <-
  dotplot(AM,
          showCategory = 10,
          title = "AM MtbAUX vs Untreated",
          font.size = font) + scale_y_discrete(
            labels = function(egoBP)
              str_wrap(egoBP, width = width)
          )

plot_grid(one, two, three, four, ncol = 2, label_size = 3)
```

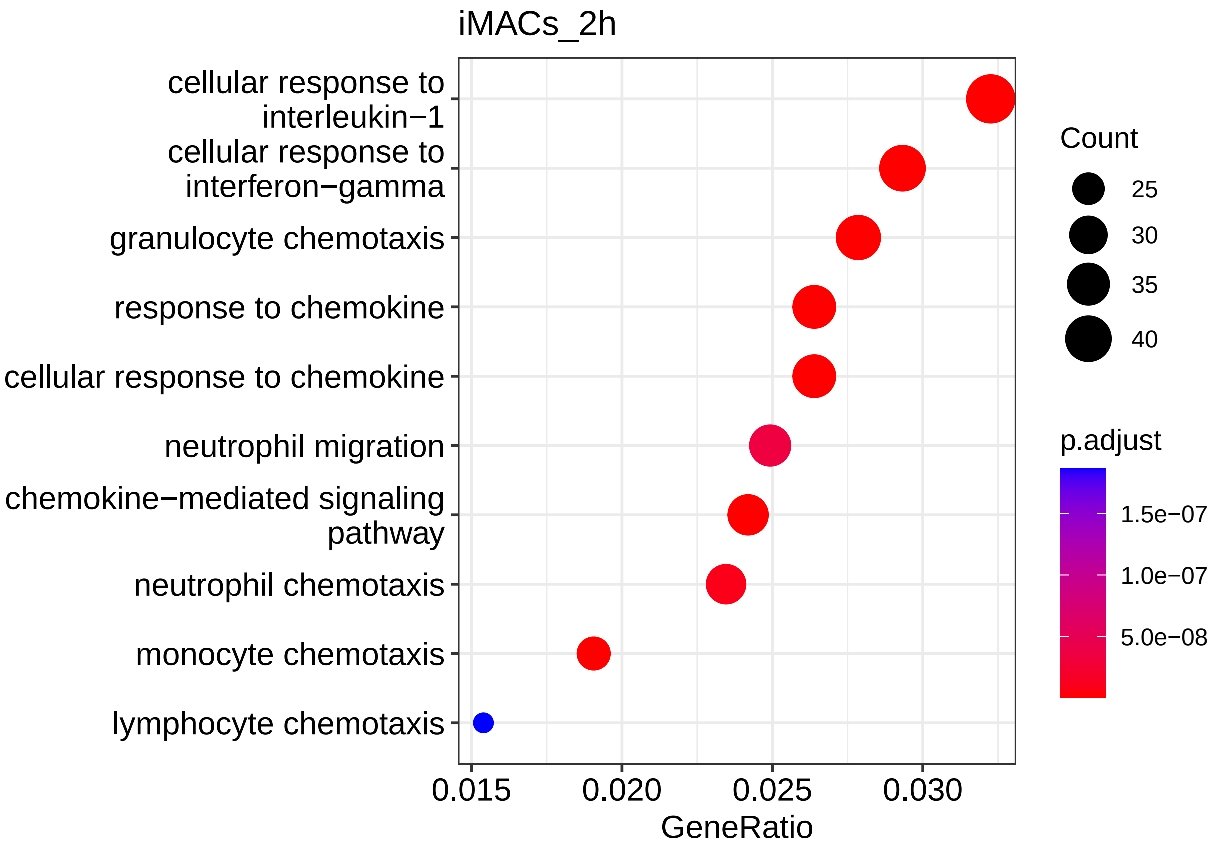


```
extract_genes_in_term <- function(enrichment_df) {
  no_terms <- 10

  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
}
```

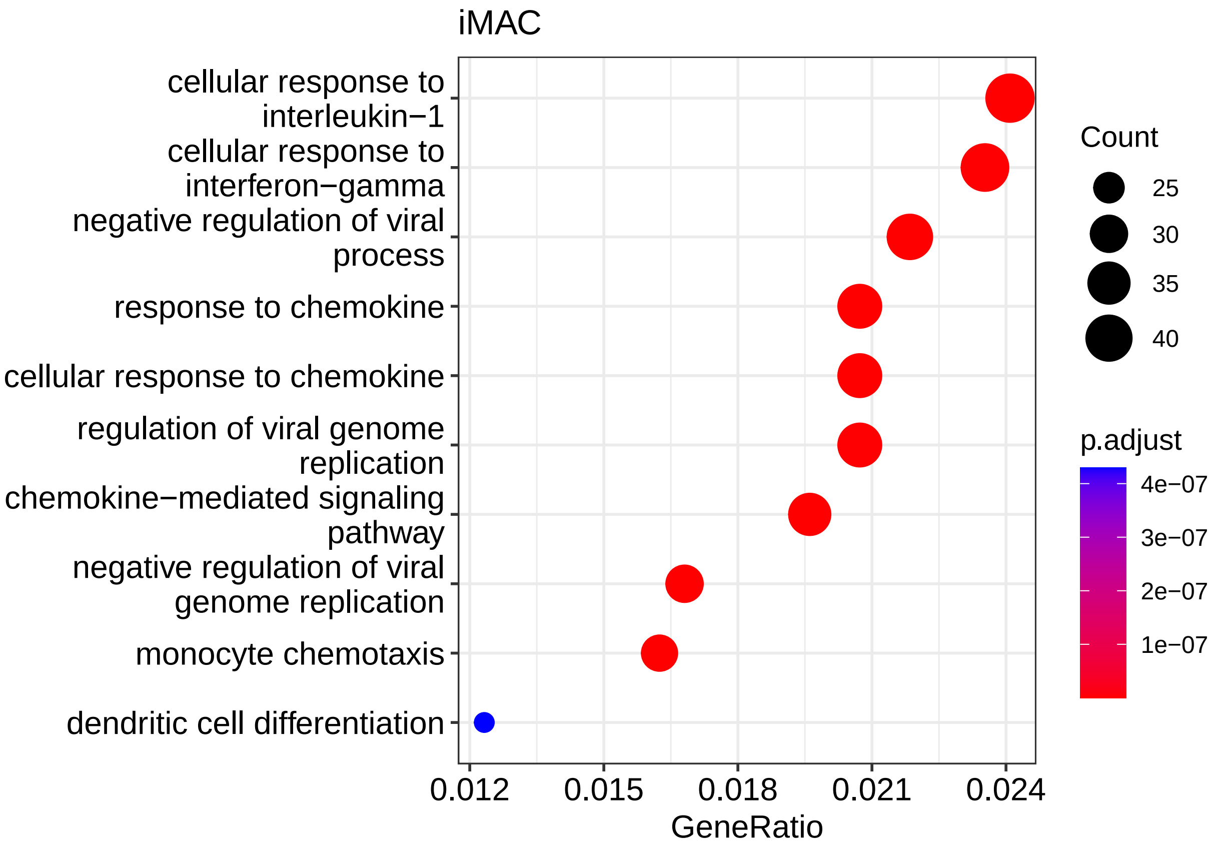
```
# extract_genes_in_term(mdm)
# extract_genes_in_term(iMAC)
# extract_genes_in_term(THP1)
# extract_genes_in_term(AM)
```

gutierrez results

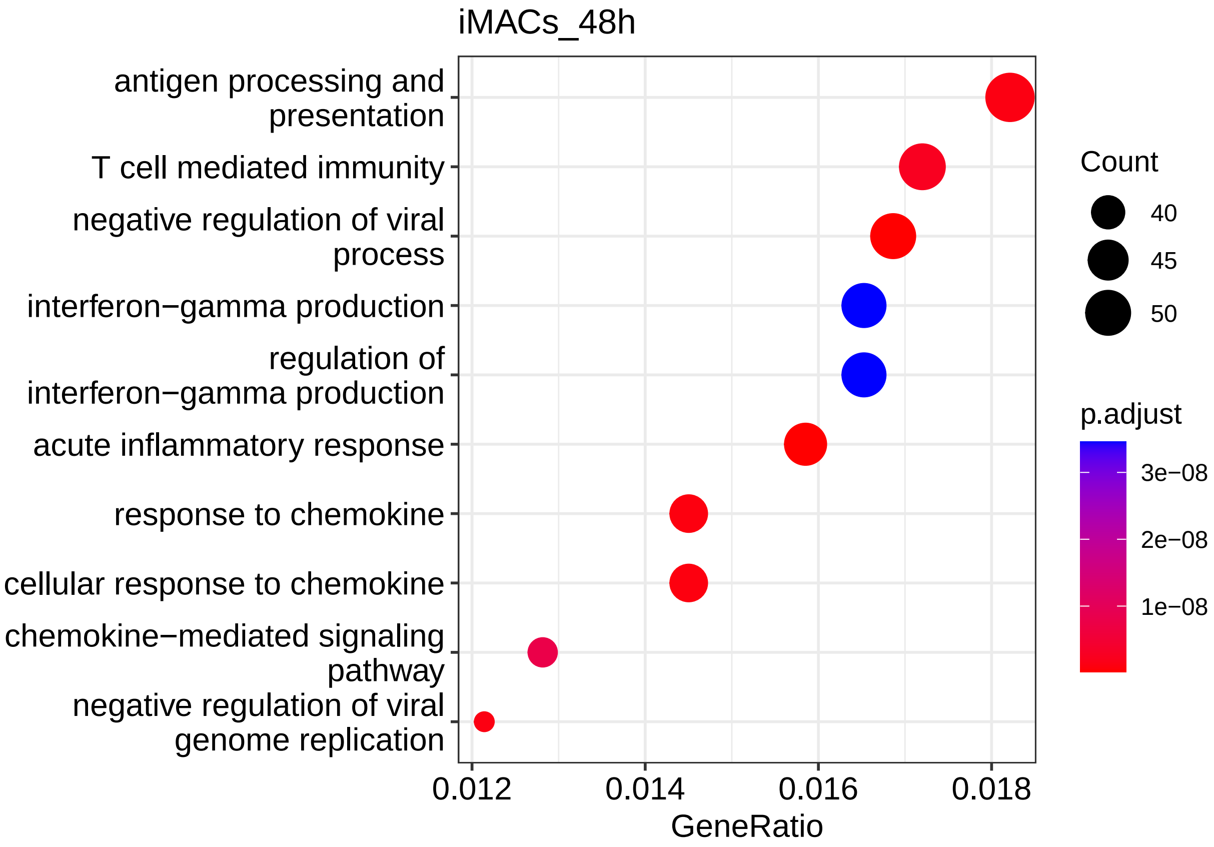
```
iMACs_2h <-
  go_enrichment(results_gut$two_hours_WTvsUninfected, adj = FALSE)

iMACs_48h <-
  go_enrichment(results_gut$fourtyeight_hours_WTvsUninfected, adj = FALSE)


dotplot(iMACs_2h, showCategory = 10, title = "iMACs_2h")
```



```
dotplot(iMAC, showCategory = 10, title = "iMAC")
```



```
dotplot(iMACs_48h, showCategory = 10, title = "iMACs_48h")
```



```r
kegg_pathview <- function(results_df, kegg_pathway, pathway_desc) {
  kegg_gene_list <- function(results_df) {
    ids <-
      clusterProfiler::bitr(
        results_df$Gene_ID,
        fromType = "ENSEMBL",
        toType = "ENTREZID",
        OrgDb = "org.Hs.eg.db"
      )
    dedup_ids = ids[!duplicated(ids[c("ENSEMBL")]), ]
    df2 = results_df[results_df$Gene_ID %in% dedup_ids$ENSEMBL, ]
    df2$Y = dedup_ids$ENTREZID
    kegg_gene_list <- df2$logFC
    kegg_gene_list

    names(kegg_gene_list) <- df2$Y
    kegg_gene_list <- na.omit(kegg_gene_list)
    kegg_gene_list = sort(kegg_gene_list, decreasing = TRUE)
    kegg_gene_list

  }

  kegg_genes <- kegg_gene_list(results_df)

  kegg_organism = "hsa"
  pathview(
    gene.data = kegg_genes,
    pathway.id = kegg_pathway,
    species = kegg_organism,
    limit = 1,
    bins = 20
  )

  new_name <-
    paste(gsub("\\..*", "", gsub("results\\$", "", deparse(
      substitute(results_df)
    ))),
    "_",
    pathway_desc,
    "_",
    "pathview",
    ".png",
    sep = "")


  file.rename(
    paste(kegg_pathway, "pathview", "png", sep = "."),
    paste("pathview_images/", new_name, sep = "")
  )

  knitr::include_graphics(paste("pathview_images/", new_name, sep = ""))

}
```

```
#Tuberculosis
Tuberculosis_pathway <- "hsa05152"

kegg_pathview(results$AM.MtbAUXvsAM.Untreated,
              Tuberculosis_pathway,
              "Tuberculosis")
kegg_pathview(results$iMACs.MtbAUXvsiMACs.Untreated,
              Tuberculosis_pathway,
              "Tuberculosis")
kegg_pathview(results$MDM.MtbAUXvsMDM.Untreated,
              Tuberculosis_pathway,
              "Tuberculosis")
kegg_pathview(results$THP1.MtbAUXvsTHP1.Untreated,
              Tuberculosis_pathway,
              "Tuberculosis")

#NFKB
NFKB_pathway <- "hsa04064"

kegg_pathview(results$AM.MtbAUXvsAM.Untreated,
              NFKB_pathway,
              "NFKB_pathway")
kegg_pathview(results$iMACs.MtbAUXvsiMACs.Untreated,
              NFKB_pathway,
              "NFKB_pathway")
kegg_pathview(results$MDM.MtbAUXvsMDM.Untreated,
              NFKB_pathway,
              "NFKB_pathway")
kegg_pathview(results$THP1.MtbAUXvsTHP1.Untreated,
              NFKB_pathway,
              "NFKB_pathway")

#TNF signalling
TNF_signalling <- "hsa04668"

kegg_pathview(results$AM.MtbAUXvsAM.Untreated,
              TNF_signalling,
              "TNF_signaling")
kegg_pathview(results$iMACs.MtbAUXvsiMACs.Untreated,
              TNF_signalling,
              "TNF_signaling")
kegg_pathview(results$MDM.MtbAUXvsMDM.Untreated,
              TNF_signalling,
              "TNF_signaling")
kegg_pathview(results$THP1.MtbAUXvsTHP1.Untreated,
              TNF_signalling,
              "TNF_signaling")

#necroptosis
Necroptosis_pathway <- "hsa04217"

kegg_pathview(results$AM.MtbAUXvsAM.Untreated,
              Necroptosis_pathway,
              "Necroptosis_pathway")
kegg_pathview(
  results$iMACs.MtbAUXvsiMACs.Untreated,
  Necroptosis_pathway,
  "Necroptosis_pathway"
)
kegg_pathview(results$MDM.MtbAUXvsMDM.Untreated,
              Necroptosis_pathway,
              "Necroptosis_pathway")
kegg_pathview(results$THP1.MtbAUXvsTHP1.Untreated,
              Necroptosis_pathway,
              "Necroptosis_pathway")


#Ferroptosis
Ferroptosis_pathway <- "hsa04216"

kegg_pathview(results$AM.MtbAUXvsAM.Untreated,
              Ferroptosis_pathway,
              "Ferroptosis_pathway")
kegg_pathview(
  results$iMACs.MtbAUXvsiMACs.Untreated,
  Ferroptosis_pathway,
  "Ferroptosis_pathway"
```

```
)
kegg_pathview(results$MDM.MtbAUXvsMDM.Untreated,
              Ferroptosis_pathway,
              "Ferroptosis_pathway")
kegg_pathview(results$THP1.MtbAUXvsTHP1.Untreated,
              Ferroptosis_pathway,
              "Ferroptosis_pathway")

#apoptosis
Apoptosis_pathway <- "hsa04210"

kegg_pathview(results$AM.MtbAUXvsAM.Untreated,
              Apoptosis_pathway,
              "Apoptosis_pathway")
kegg_pathview(results$iMACs.MtbAUXvsiMACs.Untreated,
              Apoptosis_pathway,
              "Apoptosis_pathway")
kegg_pathview(results$MDM.MtbAUXvsMDM.Untreated,
              Apoptosis_pathway,
              "Apoptosis_pathway")
kegg_pathview(results$THP1.MtbAUXvsTHP1.Untreated,
              Apoptosis_pathway,
              "Apoptosis_pathway")


# Ag processing and presentation
Antigen_pathway <- "hsa04612"

kegg_pathview(results$AM.MtbAUXvsAM.Untreated,
              Antigen_pathway,
              "Antigen_pathway")
kegg_pathview(results$iMACs.MtbAUXvsiMACs.Untreated,
              Antigen_pathway,
              "Antigen_pathway")
kegg_pathview(results$MDM.MtbAUXvsMDM.Untreated,
              Antigen_pathway,
              "Antigen_pathway")
kegg_pathview(results$THP1.MtbAUXvsTHP1.Untreated,
              Antigen_pathway,
              "Antigen_pathway")




Phagosome_maturation <- "hsa04145"

kegg_pathview(results$AM.MtbAUXvsAM.Untreated,
              Phagosome_maturation,
              "Phagosome_maturation")
kegg_pathview(
  results$iMACs.MtbAUXvsiMACs.Untreated,
  Phagosome_maturation,
  "Phagosome_maturation"
```

```
)
kegg_pathview(results$MDM.MtbAUXvsMDM.Untreated,
              Phagosome_maturation,
              "Phagosome_maturation")
kegg_pathview(
  results$THP1.MtbAUXvsTHP1.Untreated,
  Phagosome_maturation,
  "Phagosome_maturation"
)
```

within MTB treated comparison

```
Tuberculosis_pathway <- "hsa05152"

kegg_pathview(results$iMACs.MtbAUXvsAM.MtbAUX,
              Tuberculosis_pathway,
              "Tuberculosis_vs_AM")
kegg_pathview(results$MDM.MtbAUXvsAM.MtbAUX,
              Tuberculosis_pathway,
              "Tuberculosis_vs_AM")
kegg_pathview(results$THP1.MtbAUXvsAM.MtbAUX,
              Tuberculosis_pathway,
              "Tuberculosis_vs_AM")
```

```
kegg_path_enrich <- function(deg_df) {
  kegg_gene_list <- function(results_df) {
    ids <-
      clusterProfiler::bitr(
        results_df$Gene_ID,
        fromType = "ENSEMBL",
        toType = "ENTREZID",
        OrgDb = "org.Hs.eg.db"
      )
    dedup_ids = ids[!duplicated(ids[c("ENSEMBL")]), ]
    df2 = results_df[results_df$Gene_ID %in% dedup_ids$ENSEMBL, ]
    df2$Y = dedup_ids$ENTREZID
    kegg_gene_list <- df2$logFC
    kegg_gene_list

    names(kegg_gene_list) <- df2$Y
    kegg_gene_list <- na.omit(kegg_gene_list)
    kegg_gene_list = sort(kegg_gene_list, decreasing = TRUE)
    kegg_gene_list

  }

  kegg_df <- kegg_gene_list(deg_df)

  clusterProfiler::enrichKEGG(
    gene = names(kegg_df[kegg_df > 1]),
    organism = 'hsa',
    universe = names(kegg_df),
    pvalueCutoff = 0.05
    #,maxGSSize = 100
  )

}


iMAC_kegg <- kegg_path_enrich(results$iMACs.MtbAUXvsiMACs.Untreated)
AM_kegg <- kegg_path_enrich(results$AM.MtbAUXvsAM.Untreated)
MDM_kegg <- kegg_path_enrich(results$MDM.MtbAUXvsMDM.Untreated)
THP1_kegg <- kegg_path_enrich(results$THP1.MtbAUXvsTHP1.Untreated)




# dotplot(iMAC_kegg)
# dotplot(AM_kegg)
# dotplot(MDM_kegg)
# dotplot(THP1_kegg)
```

```
top10 <-
  inner_join(
    inner_join(data.frame(iMAC_kegg), data.frame(AM_kegg), by = "Description"),
    inner_join(data.frame(MDM_kegg), data.frame(THP1_kegg), by = "Description"),
    by = "Description"
  )[c(1:7, 10), 2]

top10 <-
  top10 %>% append(
    c(
      "Necroptosis",
      "Ferroptosis",
      "Pyroptosis",
      "Apoptosis",
      "Antigen processing and presentation"
    )
  )


all_kegg <-
  bind_rows(
    data.frame(iMAC_kegg, "cell" = rep("iMAC", times = nrow(
      data.frame(iMAC_kegg)
    ))),
    data.frame(AM_kegg, "cell" = rep("AM", times = nrow(
      data.frame(AM_kegg)
    ))),
    data.frame(MDM_kegg, "cell" = rep("MDM", times = nrow(
      data.frame(MDM_kegg)
    ))),
    data.frame(THP1_kegg, "cell" = rep("THP1", times = nrow(
      data.frame(THP1_kegg)
    )))
  )
```

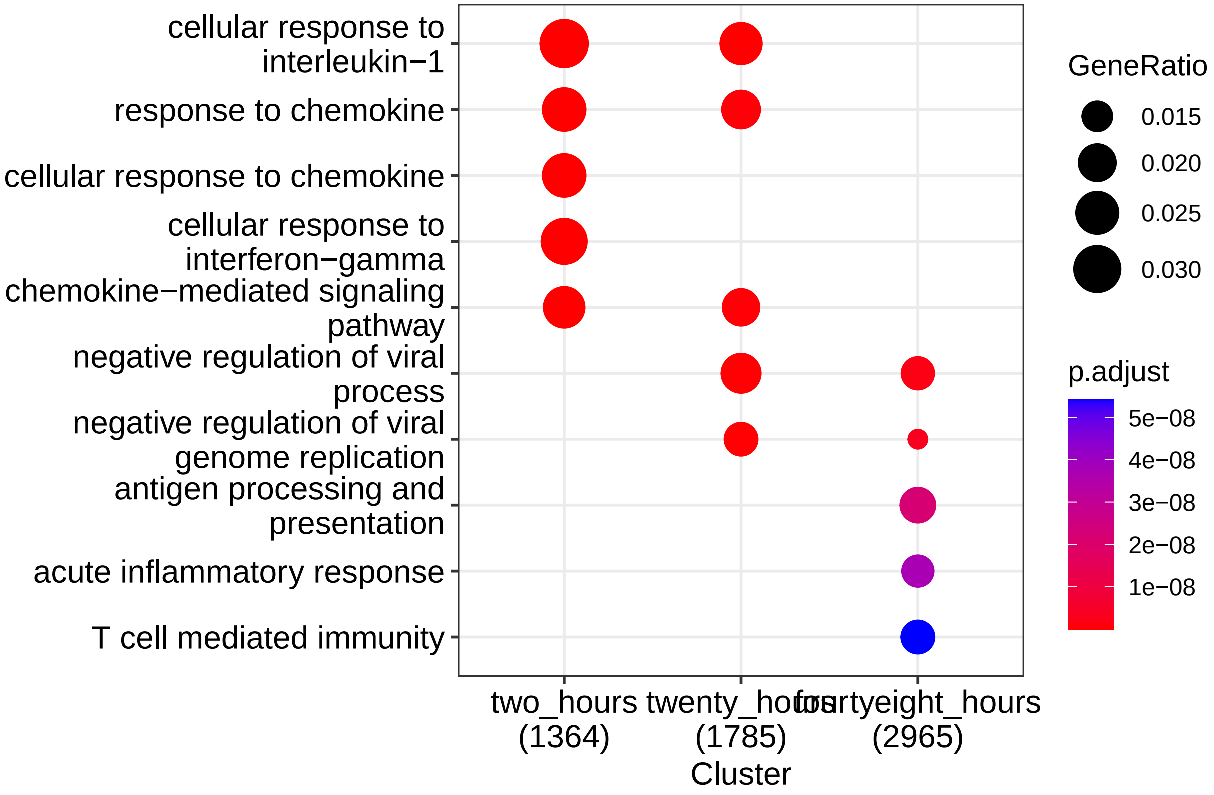
```
ggplot(all_kegg %>% filter(Description %in% top10),
       mapping = aes(Description, abs(log(qvalue)))) +
  geom_col(aes(fill = cell), position = "dodge") +
  coord_flip() +
  ggplot2::theme(axis.text.x = element_text(angle = 90))
```

```
kegg_extract <- function(kegg_df, term) {
  extract_genes_in_term <- function(enrichment_df) {
    no_terms <- length(enrichment_df$Description)

    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
  }

  necro <- extract_genes_in_term(kegg_df) %>% filter(terms == term)



  trans <-
    clusterProfiler::bitr(
      necro$gene_source,
      fromType = "ENTREZID",
      toType = "ENSEMBL",
      OrgDb = "org.Hs.eg.db"
    )
  trans <- data.frame("Gene_ID" = trans[[2]])
  trans <- trans %>% inner_join(gene.info[c(1, 11)], by = "Gene_ID")

  trans
}
```

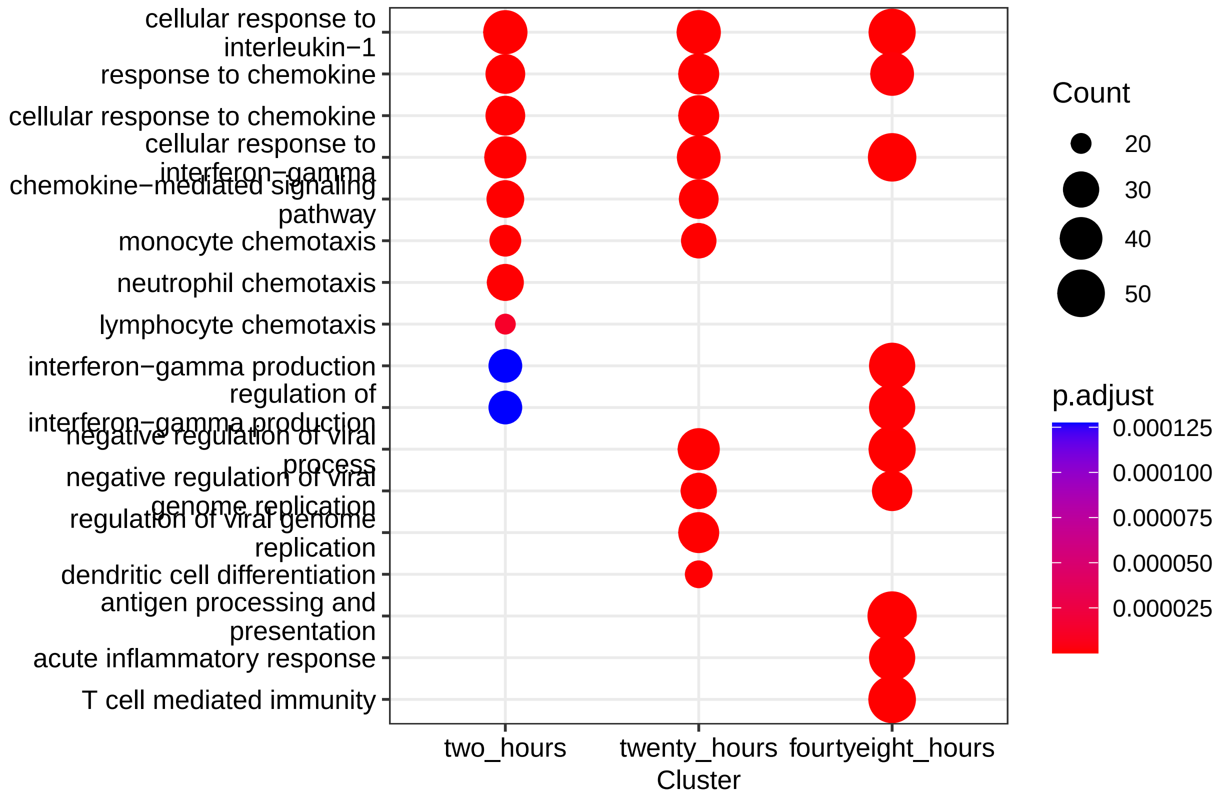
cluster compare with gut

```
two_hours <- sdeg_extraction(results_gut$two_hours_WTvsUninfected) %>% rownames_to_column("Gene_ID")
foureight_hours <-
  sdeg_extraction(results_gut$fourtyeight_hours_WTvsUninfected) %>% rownames_to_column("Gene_ID")
twenty_hours <-
  sdeg_extraction(results$iMACs.MtbAUXvsiMACs.Untreated) %>% rownames_to_column("Gene_ID")


list_iMAC <-
```

```
  list(
    "two_hours" = two_hours[[1]],
    "twenty_hours" = twenty_hours[[1]],
    "fourtyeight_hours" = foureight_hours[[1]]
  )

comp_function <- function(list) {
  enrichGO(
    gene = list,
    universe = results$iMACs.MtbAUXvsiMACs.Untreated$Gene_ID,
    keyType = "ENSEMBL",
    OrgDb = org.Hs.eg.db,
    ont = "BP",
    maxGSSize = 100,
    pAdjustMethod = "BH",
    qvalueCutoff = 0.05,
    readable = TRUE
  )

}


ck <- compareCluster(geneCluster = list_iMAC, fun = "comp_function")


dotplot(ck, includeAll = FALSE)
```



```
dotplot(
  ck,
  includeAll = FALSE,
  showCategory = 10,
  font = 10,
  size = "count"
)
```


cellular response to interleukin−1
response to chemokine
cellular response to chemokine
cellular response to interferon−gamma
chemokine−mediated signaling pathway
monocyte chemotaxis
neutrophil chemotaxis
lymphocyte chemotaxis
interferon−gamma production
regulation of interferon−gamma production
negative regulation of viral process
negative regulation of viral genome replication
regulation of viral genome replication
dendritic cell differentiation
antigen processing and presentation
acute inflammatory response
T cell mediated immunity
two_hours
twenty_hours
fourtyeight_hours
Cluster
Count
20
30
40
50
p.adjust
0.000125
0.000100
0.000075
0.000050
0.000025