GOCircle plotting of SACC-PHH RNASeq_donor treatment

Purpose:

To visualize the enriched GO terms found in the DGE profiles (donor-treatment analysis) of SACC-PHH RNASeq samples (including those sequenced in July 2018).

Load required libraries

```
library(GOplot)
```

```
## Loading required package: ggplot2

## Loading required package: ggdendro

## Loading required package: gridExtra

## Loading required package: RColorBrewer
```

```
library(gridExtra)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(tibble)
library(gage)
library(gageData)
data(kegg.gs)
data(go.sets.hs)
data(go.subs.hs)
go.bp = go.sets.hs[go.subs.hs$BP]
data(go.sets.hs)
data(go.subs.hs)
```

Load files

```
data_dir <- "Human DGEs_donortreatment/REVIGO_GO_term_output"
##All the files that were generated from REVIGO and are the condensed
##GO terms that were identified in "long form" using GAGE.
sampleFiles <- basename(Sys.glob(file.path(data_dir, "*csv")))
```

```
sampleNames <- str_replace(sampleFiles, ".csv$","")
sampleNames
```

```
## [1] "HBVvmockd8_GO_greater"   "HBVvmockd8_GO_lesser"
## [3] "coinfvHBVd28_GO_lesser"  "coinfvHBVd8_GO_lesser"
## [5] "coinfvmockd28_GO_lesser" "coinfvmockd8_GO_lesser"
```

```
data_dir_2 <- "Human DGEs_donortreatment"
sampleFiles_2 <- basename(Sys.glob(file.path(data_dir_2, "*txt")))
sampleNames_2 <- str_replace(sampleFiles_2, "^[0-9]*-*[0-9]*-*[0-9]*human_donor_treatment*","") %>%
  str_replace("_*\\s*analysis_results.txt", "")
sampleNames_2
```

```
## [1] "HumanHBVgenes-HBV_vs_mock_d28"   "HumanHBVgenes-HBV_vs_mock_d8"
## [3] "HumanHBVgenes-coinf_vs_HBV_d28"  "HumanHBVgenes-coinf_vs_HBV_d8"
## [5] "HumanHBVgenes-coinf_vs_mock_d28" "HumanHBVgenes-coinf_vs_mock_d8"
```

Functions needed

```
##Modifying the circle_dat function in GOplot
circle_dat2 <- function (terms, genes)
{
  colnames(terms) <- tolower (colnames(terms))
  terms$genes <- toupper(terms$genes)
  genes$ID <- toupper(genes$ID)
  tgenes <- strsplit(as.vector(terms$genes), ", ")
  tgenes_unlist <- unlist(tgenes) %>%
    str_replace(., "[\r\n]" , "")  ##Make sure to do this -- "\n" comes up for
                                   ##new lines when you unlist tgenes, complicating
  if (length(tgenes[[1]]) == 1)
    tgenes <- strsplit(as.vector(terms$genes), ",")
  count <- sapply(1:length(tgenes), function(x) length(tgenes[[x]]))
  logFC <- sapply(tgenes_unlist, function(x) genes$logFC[match(x,
    genes$ID)])
  logFC[is.na(logFC)] <- 0 ##change from original code so calculation of z score
                           ##can still be performed while maintaining the number of
                           ##genes falling under each GO term.
  if (class(logFC) == "factor") {
    logFC <- gsub(",", ".", gsub("\\.", "", logFC))
    logFC <- as.numeric(logFC)
  }
  s <- 1
  zsc <- c()
for (c in 1:length(count)) {
    value <- 0
    e <- s + count[c] - 1
    value <- sapply(logFC[s:e], function(x) ifelse(x > 0, 1,
                                                   ifelse(x < 0, -1 , 0)))
    value <- na.omit(value)
    zsc <- c(zsc, sum(value)/sqrt(count[c]))
    s <- e + 1
  }
  if (is.null(terms$id)) {
    df <- data.frame(category = rep(as.character(terms$category), count),
          term = rep(as.character(terms$term), count),
          count = rep(count, count),
          genes = as.character(unlist(tgenes)),
          logFC = logFC, adj_pval = rep(terms$adj_pval, count),
          zscore = rep(zsc, count), stringsAsFactors = FALSE)
  }
  else {
    df <- data.frame(category = rep(as.character(terms$category), count),
                     ID = rep(as.character(terms$id), count),
                     term = rep(as.character(terms$term), count),
                     count = rep(count, count),
                     genes = as.character(unlist(tgenes)),
                     logFC = logFC, adj_pval = rep(terms$adj_pval, count),
                     zscore = rep(zsc, count), stringsAsFactors = FALSE)
  }
  return(df)
}

##Modified version of the GOCircle function in GOplot package
GOCircle2 <- function (data, title, nsub, rad1, rad2, table.legend = T, zsc.col,
                       lfc.col, label.size, label.fontface) {
##Adding in the function for theme_blank used further down when plotting
theme_blank <- theme(axis.line = element_blank(),
                axis.text.x = element_blank(),
                axis.text.y = element_blank(),
                axis.ticks = element_blank(),
                axis.title.x = element_blank(),
                axis.title.y = element_blank(),
                panel.background = element_blank(),
                panel.border = element_blank(),
                panel.grid.major = element_blank(),
                panel.grid.minor = element_blank(),
                plot.background = element_blank())
  #Function for drawing the table in the final output along with the GO circle plot
draw_table <- function(data, col){
  id <- term <- NULL
  colnames(data) <- tolower(colnames(data))
  if (missing(col)){
    tt1 <- ttheme_default(base_size = 12)
  }else{
    text.col <- c(rep(col[1], sum(data$category == 'BP')), rep(col[2],
    sum(data$category == 'CC')), rep(col[3], sum(data$category == 'MF')))
    tt1 <- ttheme_minimal(
      core = list(bg_params = list(fill = text.col, col=NA, alpha= 1/3)),
      colhead = list(fg_params = list(col = "black")))
  }
  table <- tableGrob(subset(data, select = c(id, term)), cols = c('ID',
                    'Description'), rows = NULL, theme = tt1)
  return(table)
}

xmax <- y1 <- zscore <- y2 <- ID <- logx <- logy2 <- logy <- logFC <- NULL
  if (missing(title))
    title <- ""
```

```r
  if (missing(nsub))
    if (dim(data)[1] > 10)
      nsub <- 10
    else nsub <- dim(data)[1]
    if (missing(rad1))
      rad1 <- 2
    if (missing(rad2))
      rad2 <- 3
    if (missing(zsc.col))
      zsc.col <- c("red", "white", "blue")
    if (missing(lfc.col))
      lfc.col <- c("purple", "orange")
    else lfc.col <- rev(lfc.col)
    if (missing(label.size))
      label.size = 5
    if (missing(label.fontface))
      label.fontface = "bold"
    data$adj_pval <- -log(data$adj_pval, 10)
    suby <- data[!duplicated(data$term), ]
    if (is.numeric(nsub) == T) {
      suby <- suby[1:nsub, ]
    }
    else {
      if (strsplit(nsub[1], ":")[[1]][1] == "GO") {
        suby <- suby[suby$ID %in% nsub, ]
      }
      else {
        suby <- suby[suby$term %in% nsub, ]
      }
      nsub <- length(nsub)
    }
    N <- dim(suby)[1]
    r_pval <- adj_pval_range_10 + c(-2, 2) ##setting range of
    ##pvalues, rounded to nearest whole number.
    ##Added/subtracted two from the high and low of the range, respectively
    ymax <- c()
    for (i in 1:length(suby$adj_pval)) {
      val <- (suby$adj_pval[i] - r_pval[1])/(r_pval[2] - r_pval[1])
      ymax <- c(ymax, val)
    }
df <- data.frame(x = seq(0, 10 - (10/N), length = N), xmax = rep(10/N -  0.2, N),
                     y1 = rep(rad1, N), y2 = rep(rad2, N), ymax = ymax,
                     zscore = suby$zscore, ID = suby$ID)
    scount <- data[!duplicated(data$term), which(colnames(data) ==
                                                  "count")][1:nsub]
    idx_term <- which(!duplicated(data$term) == T)
    xm <- c()
    logs <- c()
    for (sc in 1:length(scount)) {
   idx <- c(idx_term[sc], idx_term[sc] + scount[sc] - 1)
   ##Note that val on the next line does use a magic number that may need
   ##adjustment based on the data set being used.
   val <- stats::runif(scount[sc], df$x[sc] + 0.03, (df$x[sc] + df$xmax[sc]
                                                     -0.03))
      xm <- c(xm, val)
      r_logFC <- round(range(data$logFC[idx[1]:idx[2]]), 0) + c(-1, 1)
      for (lfc in idx[1]:idx[2]) {
        val <- (data$logFC[lfc] - r_logFC[1])/(r_logFC[2] - r_logFC[1])
        logs <- c(logs, val)
      }
    }
    cols <- c()
    for (ys in 1:length(logs)) cols <- c(cols, ifelse(data$logFC[ys] > 0,
        "upregulated", ifelse(data$logFC[ys] < 0, "downregulated", "NA")))
    dfp <- data.frame(logx = xm, logy = logs, logFC = factor(cols),
                      logy2 = rep(rad2, length(logs)))
    c <- ggplot() + geom_rect(data = df, aes(xmin = x, xmax = x +  xmax,
                                             ymin = y1,
                                             ymax = y1 + ymax,
                                             fill = zscore), colour = "black") +
      geom_rect(data = df, aes(xmin = x, xmax = x + xmax, ymin = y2,
                               ymax = y2 + 1), fill = "gray90") +
      geom_rect(data = df, aes(xmin = x, xmax = x + xmax, ymin = y2 + 0.5, ymax =
          y2 +  0.5), colour = "white") +
      geom_rect(data = df, aes(xmin = x,  xmax = x +xmax, ymin = y2 + 0.25,
               ymax = y2 + 0.25),  colour = "white") +
      geom_rect(data = df, aes(xmin = x, xmax = x + xmax, ymin = y2 + 0.75,
                ymax = y2 + 0.75), colour = "white") +
      geom_text(data = df, aes(x = x + (xmax/2), y = y2 + 1.3, label = ID,
            angle = 360 - (x = x + (xmax/2))/(10/360)), size = label.size,
            fontface = label.fontface) +
      coord_polar() + labs(title = title) + ylim(1, rad2 + 1.6) + xlim(0, 10) +
      scale_fill_gradient2("z-score",  space = "Lab", low = zsc.col[3],
                           mid = zsc.col[2], high = zsc.col[1],
                           guide = guide_colourbar(title.position = "top",
                                                   title.hjust = 0.5),
                           limits = (zscore_range + c(-1, 1))) +
      theme_blank +
      theme(legend.position = "bottom",
            legend.background = element_rect(fill = "transparent"),
            legend.box = "horizontal", legend.title = element_text(size=16),
            legend.direction = "horizontal") +
      geom_point(data = dfp[which(dfp$logFC != "NA"),],
                 aes(x = logx, y = logy2 + logy),
                 pch = 21, fill = "transparent", colour = "black",
                 size = 1) + geom_point(data = dfp[which(dfp$logFC != "NA"),],
                  aes(x = logx,  y = logy2 + logy, colour = logFC), size =0.5) +
      scale_colour_manual(values = lfc.col,
      guide = guide_legend(title.position = "top", title.hjust = 0.5))
    if (table.legend) {
      table <- draw_table(suby)
      graphics::par(mar = c(0, 0.1, 0.1, 1))
      grid.arrange(c, table, ncol = 2)
    }
    else {
      c + theme(plot.background = element_rect(fill = "white"),
                panel.background = element_rect(fill = "white"))
    }
}
```

Running functions over the files of interest

```
##GO term GO:0019058 is "viral life cycle" but in go.bp it is called "viral
##infectious cycle". So changing this one term so I can match them up.
names(go.bp[3476])
```

```
## [1] "GO:0019058 viral infectious cycle"
```

```
names(go.bp)[3476] <- c("GO:0019058 viral life cycle")
names(go.bp[3476])
```

```
## [1] "GO:0019058 viral life cycle"
```

```
names(go.bp[8974])
```

```
## [1] "GO:0060337 type I interferon-mediated signaling pathway"
```

```
names(go.bp)[8974] <- c("GO:0060337 type I interferon signaling pathway")
names(go.bp[8974])
```

```
## [1] "GO:0060337 type I interferon signaling pathway"
```

```
names(go.bp[8404])
```

```
## [1] "GO:0051444 negative regulation of ubiquitin-protein ligase activity"
```

```
names(go.bp)[8404] <-
  c("GO:0051444 negative regulation of ubiquitin-protein transferase activity")

names(go.bp[4393])
```

```
## [1] "GO:0031571 mitotic cell cycle G1/S transition DNA damage checkpoint"
```

```
names(go.bp)[4393] <- c("GO:0031571 mitotic G1 DNA damage checkpoint")
names(go.bp[4393])
```

```
## [1] "GO:0031571 mitotic G1 DNA damage checkpoint"
```

```
names(go.bp[9824])
```

```
## [1] "GO:0070848 response to growth factor stimulus"
```

```
names(go.bp)[9824] <- c("GO:0070848 response to growth factor")
names(go.bp[9824])
```

```
## [1] "GO:0070848 response to growth factor"
```

```
GO_data <- function(GO_files) {
  a <- read.delim(GO_files, sep = ",")
  ##Select the columns of interest -- term_ID, description, plot_X,
  ##plot_Y, and value.
  aa <- dplyr::select(a, c(1:2, 4:5, 7)) %>%
    ##"null" in plot_X or plot_Y is "canceling" out the GO term for that row
    ##as it was considered "redudant" after running through REVIGO.
    ##So we are only keeping the rows where plot_X and plot_Y have a numeric
    ##value.
  dplyr::filter(plot_X != "null" & plot_Y != "null") %>%
    ##Adding in the required column needed by the GOplot package
  dplyr::mutate(Category = "BP") %>%
    ##Renaming and mutating columns to be in line with the input needed for GOplot
    ##package
  dplyr::rename(Term = description, ID = term_ID, adj_pval = value) %>%
  dplyr:: mutate(adj_pval=10^(adj_pval)) %>%
  dplyr::select("Category", "ID", "Term", "adj_pval") %>%
  dplyr::mutate(ID_term = paste(ID, Term, sep = " "))

GO_IDs <- aa$ID_term

  ##Slim down your go.bp to the GO terms of interest
  aaa <- go.bp[GO_IDs]
  b <- enframe(aaa)
    b$value <- gsub("\\(", '', b$value)
    b$value <- gsub("\"", '', b$value)
    b$value <- gsub("c", '', b$value)
    b$value <- gsub(")", '', b$value)
  b <- dplyr::rename(b, ID_term = name)

c <- inner_join(aa, b, by = "ID_term")

final <- dplyr::rename(c, Genes = value) %>%
  dplyr::select(Category, ID, Term, Genes, adj_pval)
##to output the order by most significant to least significant
final <- final[order(final$adj_pval),]
final

}

GO_data_output <- lapply(file.path(data_dir, sampleFiles), GO_data)
names(GO_data_output) <- sampleNames

##Assigning the range of adj_pval to a variable that you can then use for
##plotting the height of the inner circle "rectangles" of your GO circle plot
adj_pval_range <- do.call("range",sapply(GO_data_output,getElement,name="adj_pval"))
adj_pval_range
```

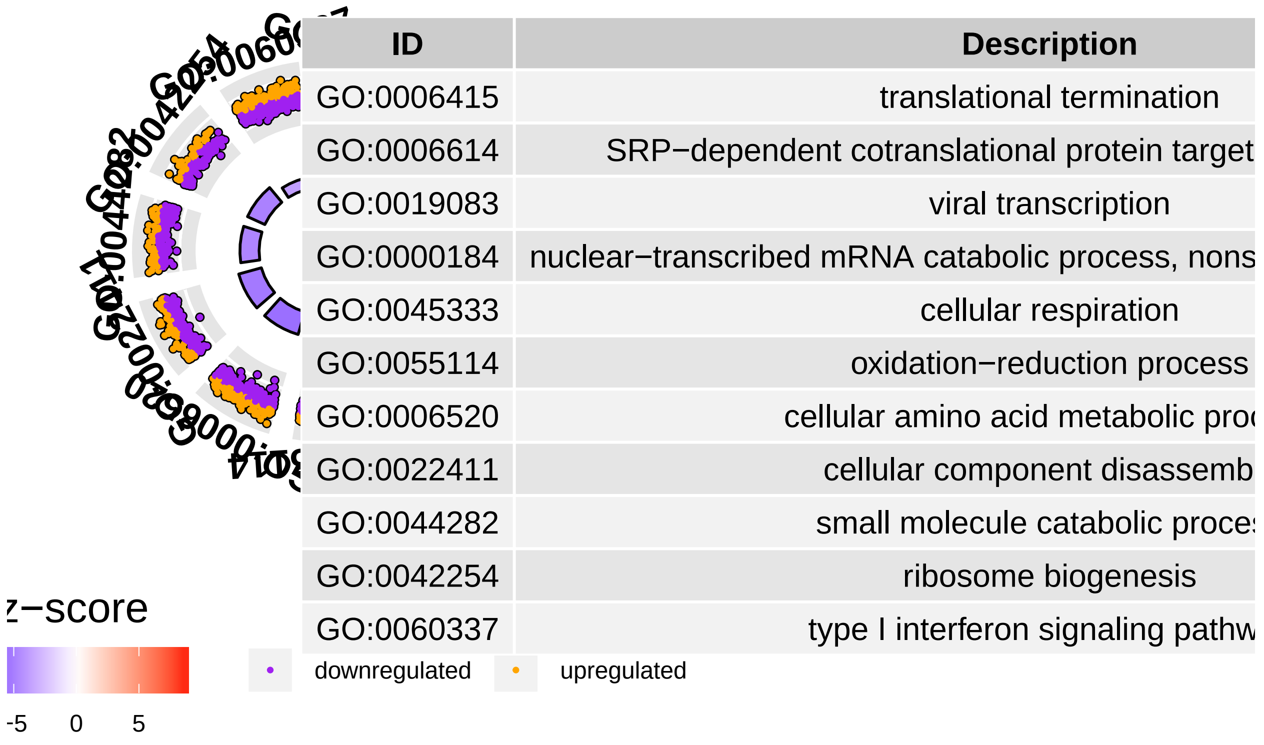
```
## [1] 1.051962e-12 4.842839e-02
```

```
adj_pval_range_10 <- rev(-log10(adj_pval_range))

DGE_data <- function(DGE_files){
d <- read.delim(DGE_files, header = TRUE)
  dd <-dplyr::select(d, log2FoldChange, padj, ENTREZID) %>%
    na.omit() %>%
    distinct(ENTREZID, .keep_all = TRUE) %>%
    dplyr::select(ENTREZID, log2FoldChange) %>%
    dplyr::rename(ID = ENTREZID, logFC = log2FoldChange)
  dd$ID <- gsub(pattern = ",.*", replacement = "", dd$ID)
  dd
}

DGE_data_output <- lapply(file.path(data_dir_2, sampleFiles_2), DGE_data)
```

```
names(DGE_data_output) <- sampleNames_2

circ_coinfvHBVd8_lesser <- circle_dat2(GO_data_output$coinfvHBVd8_GO_lesser,
DGE_data_output$`HumanHBVgenes-coinf_vs_HBV_d8`)

circ_coinfvHBVd28_lesser <-
  circle_dat2(GO_data_output$coinfvHBVd28_GO_lesser,
  DGE_data_output$`HumanHBVgenes-coinf_vs_HBV_d28`)

circ_HBVvmockd8_greater <-
  circle_dat2(GO_data_output$HBVvmockd8_GO_greater,
  DGE_data_output$`HumanHBVgenes-HBV_vs_mock_d8`)

circ_HBVvmockd8_lesser <-
  circle_dat2(GO_data_output$HBVvmockd8_GO_lesser,
  DGE_data_output$`HumanHBVgenes-HBV_vs_mock_d8`)

circ_coinfvmockd8_lesser <-
  circle_dat2(GO_data_output$coinfvmockd8_GO_lesser,
  DGE_data_output$`HumanHBVgenes-coinf_vs_mock_d8`)

circ_coinfvmockd28_lesser <-
  circle_dat2(GO_data_output$coinfvmockd28_GO_lesser,
  DGE_data_output$`HumanHBVgenes-coinf_vs_mock_d28`)

circ_list <- list(circ_coinfvHBVd28_lesser, circ_coinfvHBVd8_lesser,
                  circ_coinfvmockd28_lesser, circ_coinfvmockd8_lesser,
                  circ_HBVvmockd8_greater, circ_HBVvmockd8_lesser)


zscore_range <- do.call("range", sapply(circ_list, getElement,name="zscore"))
zscore_range
```

```
## [1] -7.966965  7.519300
```

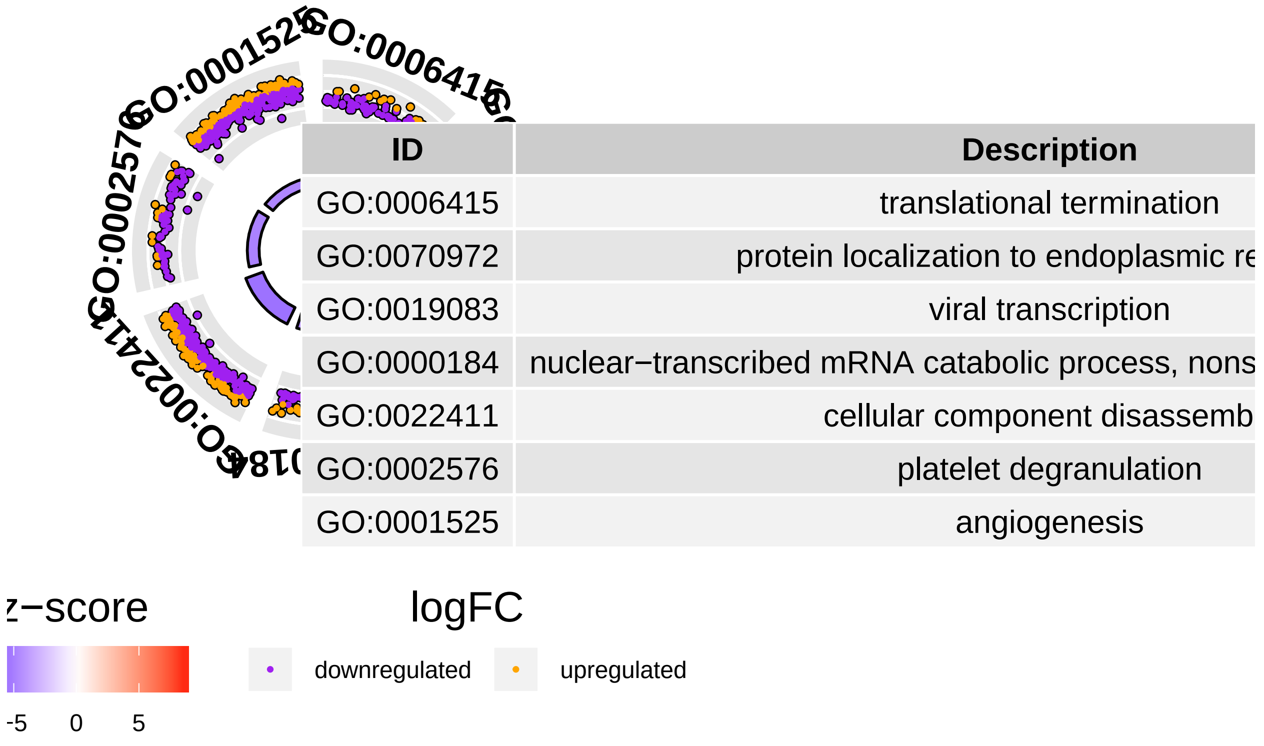
```
plot <- function(circle_dat2_output, file, nsub) {
  plotting <- GOCircle2(circle_dat2_output, nsub = nsub)
  print(plotting)
  ggsave(filename = file.path("Human DGEs_donortreatment/GO plots",
        paste(Sys.Date(), file, "plot.png")), plot = plotting,
        device = "png", width = 16, height = 8)
}

##This is to put the IFN-related GO term up on the plot
a <- GO_data_output$coinfvHBVd8_GO_lesser$ID[c(1:10, 17)] %>%
  as.character()
b <- GO_data_output$HBVvmockd8_GO_greater$ID[c(1:10, 22)] %>%
  as.character()

plot(circ_coinfvHBVd8_lesser, "coinfvHBVd8_lesser", a)
```



ID	Description
GO:0006415	translational termination
GO:0006614	SRP−dependent cotranslational protein target
GO:0019083	viral transcription
GO:0000184	nuclear−transcribed mRNA catabolic process, nons
GO:0045333	cellular respiration
GO:0055114	oxidation−reduction process
GO:0006520	cellular amino acid metabolic proc
GO:0022411	cellular component disassemb
GO:0044282	small molecule catabolic proce
GO:0042254	ribosome biogenesis
GO:0060337	type I interferon signaling pathw

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name                grob
## 1 1 (1-1,1-1) arrange      gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[colhead-fg]
```

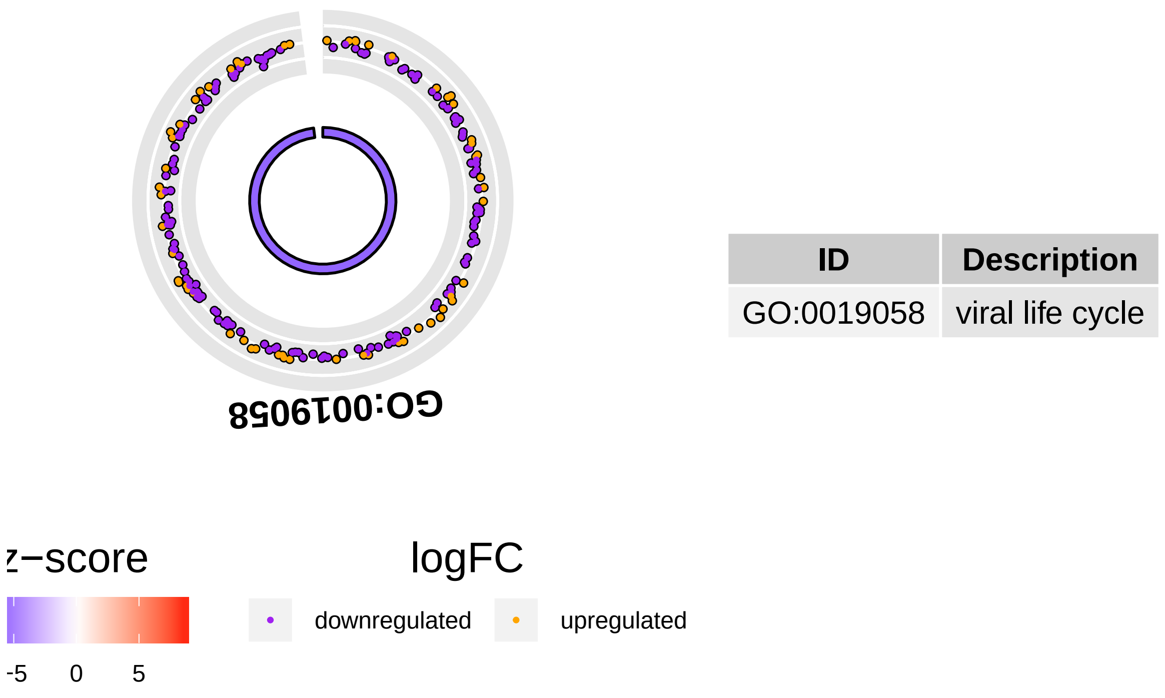
```
plot(circ_coinfvHBVd28_lesser, "coinfvHBVd28_lesser", 7)
```



ID	Description
GO:0006415	translational termination
GO:0070972	protein localization to endoplasmic re
GO:0019083	viral transcription
GO:0000184	nuclear−transcribed mRNA catabolic process, nons
GO:0022411	cellular component disassemb
GO:0002576	platelet degranulation
GO:0001525	angiogenesis

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name                 grob
## 1 1 (1-1,1-1) arrange     gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[colhead-fg]
```

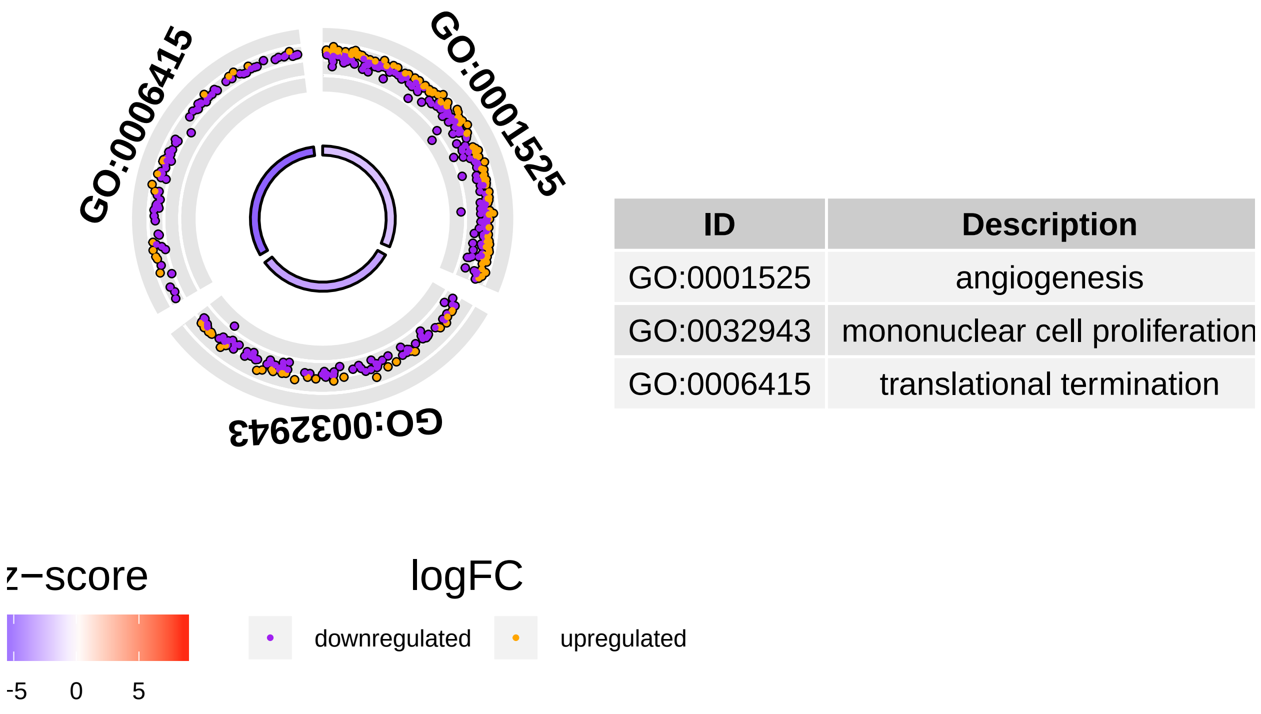
```r
plot(circ_coinfvmockd28_lesser, "coinfvmockd28_lesser", 1)
```



ID	Description
GO:0019058	viral life cycle

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name                 grob
## 1 1 (1-1,1-1) arrange     gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[colhead-fg]
```

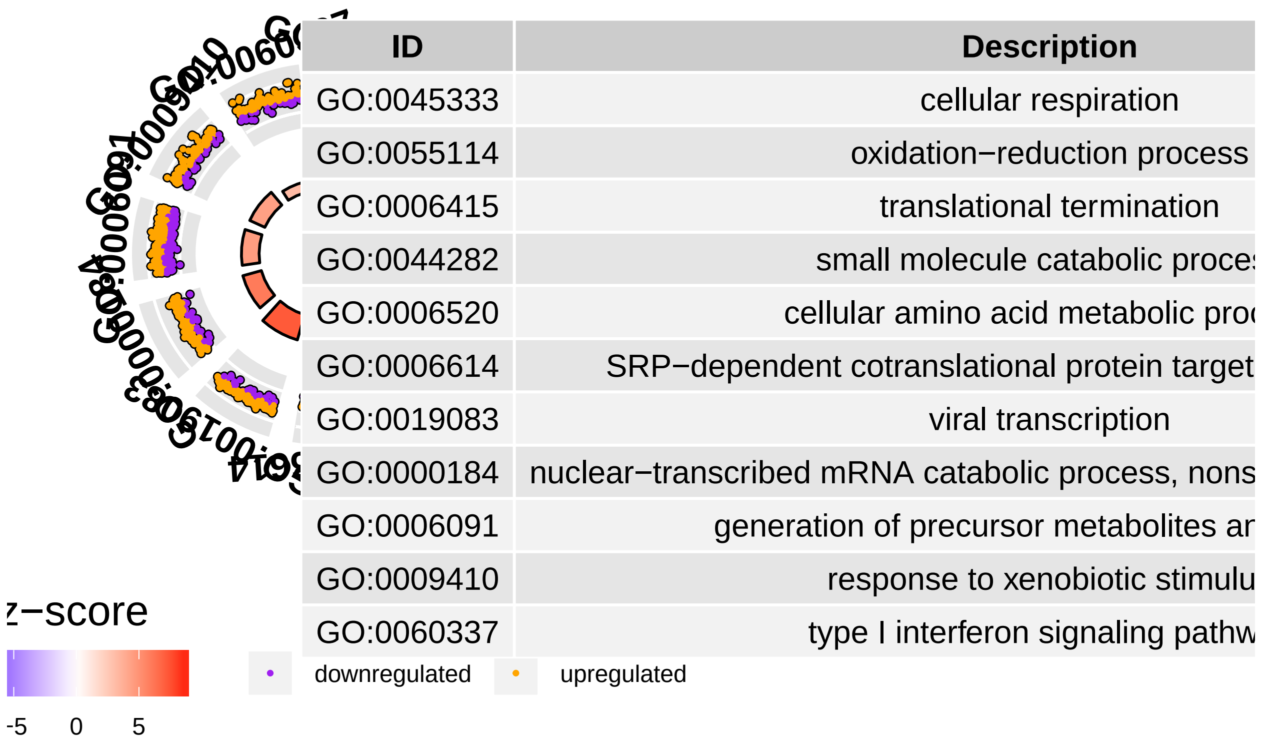
```r
plot(circ_coinfvmockd8_lesser, "coinfvmockd8_lesser", 3)
```



ID	Description
GO:0001525	angiogenesis
GO:0032943	mononuclear cell proliferation
GO:0006415	translational termination

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name                grob
## 1 1 (1-1,1-1) arrange      gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[colhead-fg]
```

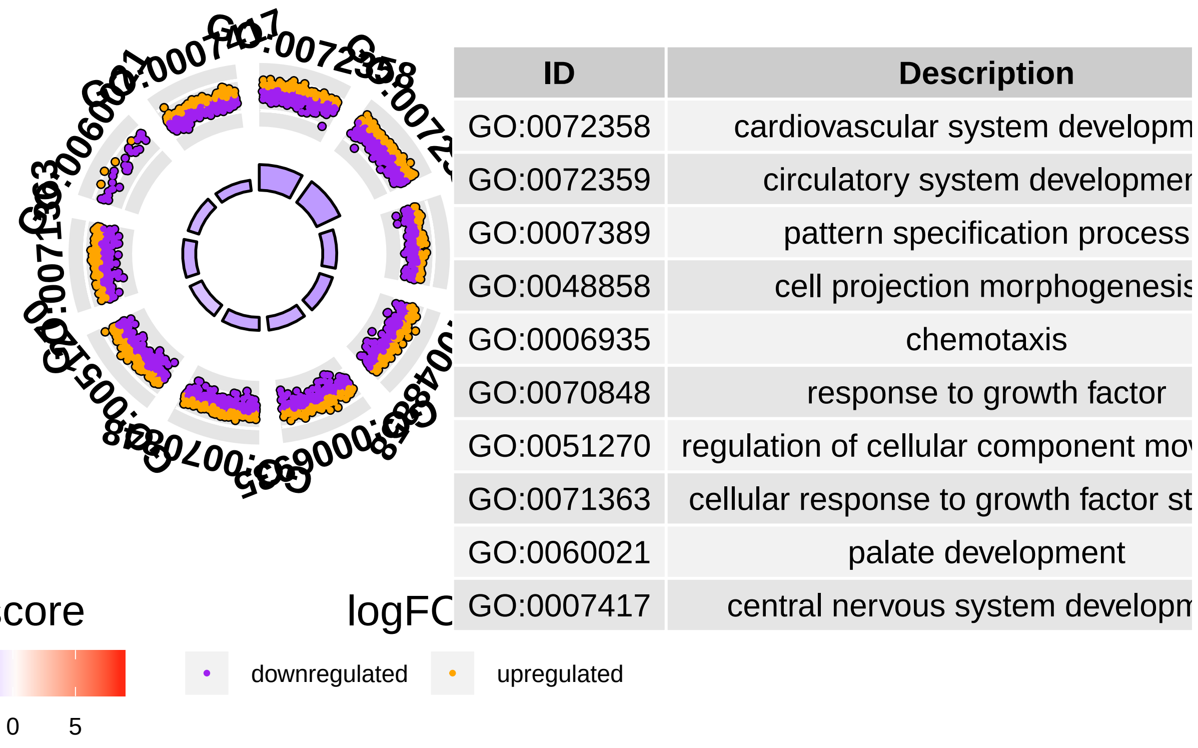
```
plot(circ_HBVvmockd8_greater, "HBVvmockd8_greater", b)
```



ID	Description
GO:0045333	cellular respiration
GO:0055114	oxidation−reduction process
GO:0006415	translational termination
GO:0044282	small molecule catabolic proce
GO:0006520	cellular amino acid metabolic proc
GO:0006614	SRP−dependent cotranslational protein target
GO:0019083	viral transcription
GO:0000184	nuclear−transcribed mRNA catabolic process, nons
GO:0006091	generation of precursor metabolites ar
GO:0009410	response to xenobiotic stimulu
GO:0060337	type I interferon signaling pathw

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name                grob
## 1 1 (1-1,1-1) arrange     gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[colhead-fg]
```

```
plot(circ_HBVvmockd8_lesser, "HBVvmockd8_lesser", 10)
```



ID	Description
GO:0072358	cardiovascular system developm
GO:0072359	circulatory system developmer
GO:0007389	pattern specification process
GO:0048858	cell projection morphogenesis
GO:0006935	chemotaxis
GO:0070848	response to growth factor
GO:0051270	regulation of cellular component mov
GO:0071363	cellular response to growth factor st
GO:0060021	palate development
GO:0007417	central nervous system developm

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name              grob
## 1 1 (1-1,1-1) arrange     gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[colhead-fg]
```

Session Info

```
sessionInfo()
```

```
## R version 3.3.3 (2017-03-06)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] bindrcpp_0.2.2     gageData_2.12.0    gage_2.24.0
##  [4] tibble_1.4.2       stringr_1.3.1      dplyr_0.7.6
##  [7] GOplot_1.0.2       RColorBrewer_1.1-2 gridExtra_2.3
## [10] ggdendro_0.1-20    ggplot2_3.0.0
##
## loaded via a namespace (and not attached):
##  [1] KEGGREST_1.14.1      tidyselect_0.2.4     purrr_0.2.5
##  [4] colorspace_1.3-2     htmltools_0.3.6      stats4_3.3.3
##  [7] yaml_2.2.0           blob_1.1.1           rlang_0.2.1
## [10] pillar_1.3.0         glue_1.3.0           withr_2.1.2
## [13] DBI_1.0.0            BiocGenerics_0.20.0  bit64_0.9-7
## [16] bindr_0.1.1          plyr_1.8.4           zlibbioc_1.20.0
## [19] Biostrings_2.42.1    munsell_0.5.0        gtable_0.2.0
## [22] evaluate_0.11        memoise_1.1.0        labeling_0.3
## [25] Biobase_2.34.0       knitr_1.20           IRanges_2.8.2
## [28] parallel_3.3.3       AnnotationDbi_1.36.2 Rcpp_0.12.18
## [31] scales_0.5.0         backports_1.1.2      S4Vectors_0.12.2
## [34] graph_1.52.0         XVector_0.14.1       bit_1.1-14
## [37] png_0.1-7            digest_0.6.15        stringi_1.2.4
## [40] grid_3.3.3           rprojroot_1.3-2      tools_3.3.3
## [43] magrittr_1.5         lazyeval_0.2.1       RSQLite_2.1.1
## [46] crayon_1.3.4         pkgconfig_2.0.1      MASS_7.3-50
## [49] assertthat_0.2.0     rmarkdown_1.10       httr_1.3.1
## [52] rstudioapi_0.7       R6_2.2.2
```