```
# ##########################################################################
# ## One-Step GO-ELITE WITH USER PARAMETERS - by Eric Dammer, Divya Nandakumar
# ##  - performs GO-Elite v1.2.3 with Fisher Exact Test for enrichment p<0.05
# ##    and 5 minimum genes per ontology
# ##
# ## Nicholas Seyfried Lab Bioinformatics - for the lab - 02/11/2019 update
# ##  - Change parameters between the lines below
# ##########################################################################
# ### GO-Elite is a python package to perform ontology analysis on gene sets.
# ### The python script for GO-Elite can be downloaded from https://www.genmapp.org/go_elite/
# ### Alternatively, there is also a GUI which can be downloaded from the same website.
# ###
# ### Custom databases can be downloaded from http://software.broadinstitute.org/gsea/msigdb/
# ### including TRANSFAC and curated pathways collected from various sources, e.g. the I2 DB.
# ### GO-Elite requires python 2.7 be installed, and PYP with command-line requires
# ### that a bugfix be applied; copy GO_Elite.py included to GO-Elite program subfolder
# ### GOEliteFolder/GO-Elite_v.1.2.0-Py/
# ##########################################################################
#
#
#
#
# setlist = ls()
# dev.off()
# options(stringsAsFactors=FALSE)
#
# ###################### EDIT THESE VARIABLES (USER PARAMETERS) ##########################################
options(stringsAsFactors=FALSE)
exps <- c("test_inputs.csv", file.path("C:", "Users", "nicholsd", "Protocols_Pipeline", "go-elite_v_projects/","test_inputs", file.path("GO-Elite_results", "CompleteResults","CRB_pruned"))

go_elite_output <- file.path("GO-Elite_results", "CompleteResults","CRB_pruned")
options(error = browser())
ls(envir=.)
##########################################################################

modsPerOntology=5
#           #Ontologies per ontology type, used for generating the PDF report; does not limit GO-Elite output

#
panelDimensions=c(3,2)
#           #rows, columns for each page of PDF output
colors=c("backgrounds","highcontrast","lightpink")
#           #colors respectively for ontologyTypes
#           #("Biological Process","Molecular Function","Cellular Component"
#           #must be valid R colors

##1. Output Report of Z-Score Bargraphs, processing all GO-Elite output files
######################## ----------------------Plotting for modules ----------------------#####################
######## this script plots the top 3 ontologies for biological process, mol function and cell component for each module


#Color scheme for ontology type key/legend (can be changed in user parameters, editing the "color" vector)
ontologyTypes=c("Biological Process","Molecular Function","Cellular Component")

modulesData <- as.list(read.csv(file.path(exps[2],exps[1]), stringsAsFactors=FALSE,header=T))
$modulesData <- as.list(read.csv(paste(exps[2],exps[1], sep="/"),sep=",", stringsAsFactors=FALSE,header=T))

listNames <- uniquemodcolors <- names(modulesData)

nModules <- length(names(modulesData))

# if(ANOVAgroups) {
#   alldata <- rbind(GOEliteforCG)
#   alldata.frame <- data.frame(Color=rep(NA,length(alldata[1])),Label=alldata)
#   uniquemodcolors <- names(GOEliteforCG) #not set above
# } else {
alldata <- uniquemodcolors #labelsInColors(c(1:nModules)
alldata1c <- paste("MT",seq(1:nModules),sep="")
alldata.frame <- as.data.frame(data.frame(Color=alldata,Label=paste0(alldata1c)," ",alldata1c))
#}

setwd(paste(file.path(exps[2],exps[3])))
pdf(paste0("GO-Elite_",exps[3],"_pdf",),height=10,width=8)
op <- par(mfrow=panelDimensions,oma=c(0,0,3,0))
frame()
legend(x="topleft",legend = ontologyTypes, fill=colors, title=" ",cex=2,horiz=F,xpd=T)
legend(x="topleft",legend = c(" "," "," "), title="Ontology Types",cex=2.5,horiz=F,xpd=T, bty='n', title.adj=1.4)
#frame()
GOEliteOUTFileTrailer= c("-GO_z-score_elite.txt")
summary <- list()
for(i in 1:length(uniquemodcolors)){{{
  listmodcolor=uniquemodcolors[i]
  if (file.exists(file.path(exps[2],exps[3],go_elite_output,paste(thismod,GOEliteOUTFileTrailer, sep=""))) == F) { #"note "_Module" needs to be removed from fileTrailer if setting uniquemodcolors not by module color
    if(file.exists(file.path(exps[2],exps[3],go_elite_output, paste(thismod,"_2",GOEliteOUTFileTrailer, sep=""))) == T) {
      tmp=read.csv(file=file.path(exps[2],exps[3],go_elite_output,paste(thismod,"_2",GOEliteOUTFileTrailer, sep="")),sep="\t")
    } else {
      next
    }
  } else {
    tmp=read.csv(file=file.path(exps[2],exps[3],go_elite_output,paste(thismod,GOEliteOUTFileTrailer, sep="")),sep="\t") #"note "_Module" needs to be removed from fileTrailer if setting uniquemodcolors not by module color
  }

  if (length(tmp[,1]) == 0) next
  tmp = tmp[,c(2,3,5,10,11)] ## Select GO-terms,GO-Type, Z-score,pValues and gene lists
  tmp1 = tmp[order(tmp[Z.Score,decreasing=T),]
  tmp2 = tmp[order(tmp[GOOntology.Type,decreasing=T),] #sex tmp2
  tmp2 = tmp[tmp[GOOntology.Type == "biological process",][c(1:numBestPerOntology),]
  tmp2 = rbind(tmp2,tmp[tmp[GOOntology.Type == "molecular function",][c(1:numBestPerOntology),] )
  tmp2 = rbind(tmp2,tmp[tmp[GOOntology.Type == "cellular component",][c(1:numBestPerOntology),] )
  tmp2 <- na.omit(tmp2)
#     tmp2 <- tmp2[order(tmp2[Z.Score,decreasing=T),] #added this row, if you want to mix ontology types and sort by Z.Score only
  tmp2 <- tmp2[rev(rownames(tmp2)),]

  summary[[i]] <- tmp2

  ### To color bars by mol function, cell component or biological process
  for (j in 1:nrow(tmp2)){
    if (tmp2[GOOntology.Type[j] == "molecular function"){
      tmp2Color[j] <- color[2]
    } else if (tmp2[GOOntology.Type[j] == "cellular component"){
      tmp2Color[j] <- color[3]
    } else if (tmp2[GOOntology.Type[j] == "biological process"){
      tmp2Color[j] <- color[1]
    }
    # tmp2Color[j] <- uniquemodcolors[i] #module color for all bars, instead of different colors by ontology type
  }

  if (tmp2[Z.Score == F) next
  par(mar=c(12,12,4,3))
  xlim <- c(0,1.1*max(tmp2[Z.Score))
  moduleTitle <- alldata.frame[i,"Label"]
  ab <- barplot(tmp2[Z.Score,horiz = TRUE,width =0.85,lwd=1,main=moduleTitle, xlim=xlim,col=tmp2Color,cex.axis=2.7,xlab="Z Score",cex.lab=3.5,cex.main=3.55,ylim=c(0,nrow(tmp2)+0.8))
  abline(v=1.96,col="red", cex.main = 0.5)
  axis(2, at=ab, labels = tmp2[GOOntology.Name, tick=FALSE, las =2, line =-0.5, cex.main = 0.7)
}

par(op) # Leaves the last plot
dev.off()
```