04-Enrichment

R Markdown - Enrichment anlalysis using EnrichR R package

Summary - Input gene lists created from T-Test / EdgeR group comparison analysis . Or an short list of genes of interest from your collaborator (Researcher/Clinician/) - If needed, short list genes based on p-value or FDR value. - Run EnrichR on short listed gene list

Import T-test results so that we can short list further

```
#folder that contains group comparison results
fileName <- "input/CRCPilot_Ttest_Shortlisted.csv"

# read in group comparison results
result1 <- read.csv(file = fileName,
                        header = T, stringsAsFactors = F, row.names = 1)
```

#Filtering/shortlisting example - 1 Short list results based in p-value cut off (pvalue <= 0.05)

```
#pValueCutOff <- 0.05
#which1 <- ((as.numeric(result1$Pvalue) <= pValueCutOff))
#table(which1) #146 genes with p <= 0.01. 719 genes with p-value <= 0.05

#resultShort <- result1[which1, ] #short listed genes
```

#Filtering/shortlisting example - 2 Short list results based in p-value cut off (pvalue <= 0.01)

```
pValueCutOff <- 0.01
which1 <- ((as.numeric(result1$Pvalue) <= pValueCutOff))
table(which1) #146 genes with p <= 0.01.
```

```
## which1
## FALSE  TRUE
##   573   146
```

```
resultShort <- result1[which1, ] #short listed genes
```

#Filtering/shortlisting example - 3 Short list results based in p-value cut off (p-value <= 0.01), and fold change cut off (greater than 1.5 or less than -1.5),

```
#pValueCutOff <- 0.01
#which1 <- ((as.numeric(result1$Pvalue) <= pValueCutOff) & (as.numeric(result1$SignedFC) >= 1.5 | as.nu
#table(which1) #26 genes TRUE

#resultShort <- result1[which1, ] #short listed genes
```

#Filtering/shortlisting example - 4 Sort by FDR . Take top 500 genes

```
#resultOrder <- result1[order(result1$FDR),]
#resultShort <- resultOrder[1:500, ]
```

#Filtering/shortlisting example - 5 Short list results based in adjusted p-value cut off (also called q-value or fdr) <= 0.05

```
#qValueCutOff <- 0.05
#which1 <- ((as.numeric(result1$FDR) <= qValueCutOff))
#table(which1) #
#resultShort <- result1[which1, ] #short listed genes
```

Export the short listed results for reference

Make sure folder "output" exists

```
#length of short listed results
nrow(resultShort)
```

```
## [1] 146
```

```
write.csv(x = resultShort, file = "output/TTest_results_shortlist2.csv")
```

Clean gene names

Note - the gene names are in the form of "reporterid_genename". So need to split this.

```
funcSplit <- function(rep_gene) {
    rep_gene_split <- unlist(strsplit(x = rep_gene,
                                      split = "_",
                                      fixed = TRUE))
    gene <- rep_gene_split[2]
    return(gene)
}
geneListSplit <- apply(X = as.matrix(row.names(resultShort)),
                       MARGIN = 1, FUN = funcSplit )
head(geneListSplit) #cleaned gene names
```

```
## [1] "LOC150197" "UBD"       "SAMSN1"    "TMEM81"    "PARP15"    "THNSL2"
```

```
#remove duplicates
geneListSplit1 <- unique(geneListSplit)

# remove NA value
geneListFinal <- na.omit(geneListSplit1) #701 genes

#print number of unique genes
length(geneListFinal)
```

```
## [1] 140
```

```
write.table(x = geneListFinal,
          file = "output/shortListedUniqueGenes.tsv",
          quote = F, sep = "\t", row.names = FALSE, col.names = F)
```

Load Databases for Enrichr R package , and check connection

```
#checking if EnrichR website and packing are working
#dbs <- enrichR::listEnrichrDbs()  #total number of databases available = 200+

#testing if EnrichR package is working
testOutput <- enrichR::enrichr(genes = c("Runx1", "Gfi1", "Gfi1b", "Spi1", "Gata1", "Kdr"), databases =
```

```
## Uploading data to Enrichr... Done.
##   Querying KEGG_2021_Human... Done.
## Parsing results... Done.
```

```
head(testOutput[[1]])
```

```
##                                     Term Overlap      P.value Adjusted.P.value
## 1                  Acute myeloid leukemia    2/67 0.0001643951      0.002794717
## 2 Transcriptional misregulation in cancer   2/192 0.0013407651      0.011396503
## 3                      Pathways in cancer   2/531 0.0098313553      0.055711013
## 4                  VEGF signaling pathway    1/59 0.0175720140      0.074681059
## 5                Chronic myeloid leukemia    1/76 0.0225871297      0.076796241
## 6               Th17 cell differentiation   1/107 0.0316774286      0.079200112
##   Old.P.value Old.Adjusted.P.value Odds.Ratio Combined.Score      Genes
## 1           0                     0  153.30000     1335.73937 SPI1;RUNX1
## 2           0                     0   52.11579      344.72067 SPI1;RUNX1
## 3           0                     0   18.39792       85.03847 SPI1;RUNX1
## 4           0                     0   68.74483      277.82863        KDR
## 5           0                     0   53.11733      201.33461      RUNX1
## 6           0                     0   37.52453      129.54033      RUNX1
```

```
#List of databases for which enrichment analysis will be run
dblist1 <- read.csv(file = "input/2023-EnrichR-Databases.txt",
                    header = F, stringsAsFactors = F)

head(dblist1)
```

```
##                           V1
## 1            KEGG_2021_Human
## 2     WikiPathway_2021_Human
## 3 GO_Biological_Process_2023
## 4              Reactome_2022
## 5             BioPlanet_2019
## 6               ClinVar_2019
```

call function to run Enrichment

```
# set output file name
outputFileName1 <- paste("output/ColonCancer", "_EnrichR.xlsx", sep="")

#Load R script into the environment
source(file = "functionEnrichment.R")

#call function to run Enrichment
functionEnrichment(dblist1, geneListFinal, outputFileName1)
```

```
## Uploading data to Enrichr...
```

```
## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.
```

```
## Done.
##   Querying KEGG_2021_Human... Done.
## Parsing results... Done.
## Uploading data to Enrichr...
```

```
## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
```

```
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying WikiPathway_2021_Human... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying GO_Biological_Process_2023... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying Reactome_2022... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying BioPlanet_2019... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying ClinVar_2019... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying Transcription_Factor_PPIs... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying TRANSFAC_and_JASPAR_PWMs... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying TargetScan_microRNA... Done.
## Parsing results... Done.
```

```
## Uploading data to Enrichr...
## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.
## Done.
##   Querying miRTarBase_2017... Done.
## Parsing results... Done.
## Uploading data to Enrichr...
## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.
## Done.
##   Querying DisGeNET... Done.
## Parsing results... Done.
## Uploading data to Enrichr...
## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.
## Done.
##   Querying OMIM_Disease... Done.
## Parsing results... Done.
## Uploading data to Enrichr...
## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.
## Done.
##   Querying Jensen_DISEASES... Done.
## Parsing results... Done.
## Uploading data to Enrichr...
## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.
## Done.
##   Querying Chromosome_Location... Done.
## Parsing results... Done.
## Uploading data to Enrichr...
## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.
## Done.
##   Querying VirusMINT... Done.
## Parsing results... Done.
## Uploading data to Enrichr...
## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.
## Done.
##   Querying Virus-Host_PPI_P-HIPSTer_2020... Done.
## Parsing results... Done.
## Uploading data to Enrichr...
## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.
```

```
## Done.
##   Querying HMDB_Metabolites... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying dbGap... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying MSigDB_Hallmark_2020... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying ProteomicsDB_2020... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying GWAS_Catalog_2023... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying InterPro_Domains_2019... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying CCLE_Proteomics_2020... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying Proteomics_Drug_Atlas_2023... Done.
## Parsing results... Done.
## Uploading data to Enrichr...
```

```
## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying PheWeb_2019... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying Pfam_Domains_2019... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying ChEA_2022... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying SILAC_Phosphoproteomics... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying Azimuth_2023... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying MAGNET_2023... Done.
## Parsing results... Done.
## Uploading data to Enrichr...

## Warning in enrichR::enrichr(unlist(genes1), oneDB): genes must be a non-empty
## vector of gene names or a data.frame with genes and score.

## Done.
##   Querying GeDiPNet_2023... Done.
## Parsing results... Done.
#NEED INTERNET CONNECTION
```

Note - you will need internet connection to complete the above step.

RESULT

When you run the funtion, will create an Excel file with multiple tabs. Each tab shows the enrichment results for a database. A total of 14-15 databases, so 14-15 tabs in the excel sheet

After you finish coding, click on “Knit” to create an HTML or PDF file with text and code