Gene set enrichment Proximity-labeling Proteomics

2023-11-05

This script is for conducting gene set enrichment analysis (GSEA) on proximity-labeling proteomics datasets. It identifies biological processes and pathways enriched among differentially expressed genes in a dataset. Key steps include installing necessary libraries, loading gene expression data, preparing gene lists, and running enrichment analyses using the Gene Ontology (GO). Visualization options, such as dot plots, help to interpret which gene sets are most relevant to the biological questions in the study.

Based on the following script: https://learn.gencore.bio.nyu.edu/rna-seq-analysis/gene-set-enrichment-analysis/

Install packages

```r
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
```

```
## Bioconductor version '3.16' is out-of-date; the current release version '3.20'
##   is available with R version '4.4'; see https://bioconductor.org/install
```

```r
BiocManager::install("clusterProfiler")
```

```
## Bioconductor version 3.16 (BiocManager 1.30.22), R 4.2.2 (2022-10-31)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   `force = TRUE` to re-install: 'clusterProfiler'
```

```
## Old packages: 'ape', 'aplot', 'askpass', 'backports', 'BH', 'BiocManager',
##   'bit', 'bit64', 'bitops', 'boot', 'brew', 'brio', 'broom', 'bslib', 'cachem',
##   'callr', 'cli', 'cluster', 'codetools', 'colorspace', 'commonmark',
##   'cowplot', 'cpp11', 'crayon', 'credentials', 'curl', 'data.table', 'DBI',
##   'dbplyr', 'dendextend', 'desc', 'digest', 'downlit', 'evaluate', 'fansi',
##   'farver', 'fastmap', 'foreign', 'fs', 'gert', 'ggforce', 'ggfun', 'ggh4x',
##   'ggnewscale', 'ggplot2', 'ggraph', 'ggrepel', 'gh', 'glue', 'graphlayouts',
##   'gtable', 'haven', 'highr', 'htmltools', 'htmlwidgets', 'httpuv', 'httr2',
##   'igraph', 'jsonlite', 'KernSmooth', 'knitr', 'later', 'lattice', 'markdown',
##   'mgcv', 'munsell', 'nlme', 'openssl', 'patchwork', 'pkgbuild', 'pkgdown',
##   'pkgload', 'plotly', 'polyclip', 'processx', 'profvis', 'progress',
##   'promises', 'ps', 'ragg', 'Rcpp', 'RcppArmadillo', 'RcppEigen', 'RCurl',
##   'readr', 'remotes', 'reprex', 'rlang', 'rmarkdown', 'roxygen2', 'rpart',
##   'RSQLite', 'rstudioapi', 'RUnit', 'rvest', 'sass', 'scales', 'scatterpie',
##   'seriation', 'shadowtext', 'shiny', 'stringi', 'survival', 'sys',
##   'systemfonts', 'testthat', 'textshaping', 'tidygraph', 'tidyr', 'tidyselect',
##   'tidytree', 'timechange', 'tinytex', 'tweenr', 'usethis', 'uuid', 'vctrs',
##   'vegan', 'viridis', 'vroom', 'waldo', 'withr', 'xfun', 'XML', 'xml2',
##   'xopen', 'yaml', 'yulab.utils', 'zip'
```

#install DOSE

```r
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("DOSE")
```

```
## Bioconductor version 3.16 (BiocManager 1.30.22), R 4.2.2 (2022-10-31)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##    `force = TRUE` to re-install: 'DOSE'

## Old packages: 'ape', 'aplot', 'askpass', 'backports', 'BH', 'BiocManager',
##    'bit', 'bit64', 'bitops', 'boot', 'brew', 'brio', 'broom', 'bslib', 'cachem',
##    'callr', 'cli', 'cluster', 'codetools', 'colorspace', 'commonmark',
##    'cowplot', 'cpp11', 'crayon', 'credentials', 'curl', 'data.table', 'DBI',
##    'dbplyr', 'dendextend', 'desc', 'digest', 'downlit', 'evaluate', 'fansi',
##    'farver', 'fastmap', 'foreign', 'fs', 'gert', 'ggforce', 'ggfun', 'ggh4x',
##    'ggnewscale', 'ggplot2', 'ggraph', 'ggrepel', 'gh', 'glue', 'graphlayouts',
##    'gtable', 'haven', 'highr', 'htmltools', 'htmlwidgets', 'httpuv', 'httr2',
##    'igraph', 'jsonlite', 'KernSmooth', 'knitr', 'later', 'lattice', 'markdown',
##    'mgcv', 'munsell', 'nlme', 'openssl', 'patchwork', 'pkgbuild', 'pkgdown',
##    'pkgload', 'plotly', 'polyclip', 'processx', 'profvis', 'progress',
##    'promises', 'ps', 'ragg', 'Rcpp', 'RcppArmadillo', 'RcppEigen', 'RCurl',
##    'readr', 'remotes', 'reprex', 'rlang', 'rmarkdown', 'roxygen2', 'rpart',
##    'RSQLite', 'rstudioapi', 'RUnit', 'rvest', 'sass', 'scales', 'scatterpie',
##    'seriation', 'shadowtext', 'shiny', 'stringi', 'survival', 'sys',
##    'systemfonts', 'testthat', 'textshaping', 'tidygraph', 'tidyr', 'tidyselect',
##    'tidytree', 'timechange', 'tinytex', 'tweenr', 'usethis', 'uuid', 'vctrs',
##    'vegan', 'viridis', 'vroom', 'waldo', 'withr', 'xfun', 'XML', 'xml2',
##    'xopen', 'yaml', 'yulab.utils', 'zip'
```

#If installed all start here

Load libraries

```r
library(clusterProfiler)
```

```
##

## Registered S3 methods overwritten by 'treeio':
##    method             from
##    MRCA.phylo         tidytree
##    MRCA.treedata      tidytree
##    Nnode.treedata     tidytree
##    Ntip.treedata      tidytree
##    ancestor.phylo     tidytree
##    ancestor.treedata  tidytree
##    child.phylo        tidytree
##    child.treedata     tidytree
##    full_join.phylo    tidytree
##    full_join.treedata tidytree
##    groupClade.phylo   tidytree
##    groupClade.treedata tidytree
##    groupOTU.phylo     tidytree
##    groupOTU.treedata  tidytree
##    is.rooted.treedata tidytree
##    nodeid.phylo       tidytree
##    nodeid.treedata    tidytree
##    nodelab.phylo      tidytree
##    nodelab.treedata   tidytree
##    offspring.phylo    tidytree
##    offspring.treedata tidytree
```

```
##    parent.phylo        tidytree
##    parent.treedata     tidytree
##    root.treedata       tidytree
##    rootnode.phylo      tidytree
##    sibling.phylo       tidytree
```

```
## clusterProfiler v4.6.2  For help: https://yulab-smu.top/biomedical-knowledge-mining-book/
##
## If you use clusterProfiler in published research, please cite:
## T Wu, E Hu, S Xu, M Chen, P Guo, Z Dai, T Feng, L Zhou, W Tang, L Zhan, X Fu, S Liu, X Bo, and G Yu.
```

```
##
## Attaching package: 'clusterProfiler'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```r
library(ggplot2)
```

Load Organism Annotation Data

```r
organism = "org.Hs.eg.db"
library(organism, character.only = TRUE)
```

```
## Loading required package: AnnotationDbi
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

```
## Loading required package: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
## Loading required package: IRanges
```

```
## Loading required package: S4Vectors
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:clusterProfiler':
##
##     rename

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:clusterProfiler':
##
##     slice

##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:clusterProfiler':
##
##     select

##
```

Read proximity-proteomics dataset

```r
df<-read.csv("/Users/jeffreyreina/Documents/Salk/TurboID analysis/MDATurboUniprot.csv",header=TRUE,row.
```

Prepare Gene List for Enrichment Analysis

```r
df$Uniprot<-rownames(df) ## Create a new column for Uniprot IDs
```

Organize data frame

```r
# Extract log2 fold change values
original_gene_list <- df$log2FC
names(original_gene_list) <- df$Uniprot

# Remove any NA values and sort by decreasing order
gene_list<-na.omit(original_gene_list)
gene_list = sort(gene_list, decreasing = TRUE)
```

Check Available Key Types

```r
keytypes(org.Hs.eg.db) ##we will use UNIPROT
```

```
##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
##  [6] "ENTREZID"     "ENZYME"       "EVIDENCE"     "EVIDENCEALL"  "GENENAME"
## [11] "GENETYPE"     "GO"           "GOALL"        "IPI"          "MAP"
## [16] "OMIM"         "ONTOLOGY"     "ONTOLOGYALL"  "PATH"         "PFAM"
## [21] "PMID"         "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"
## [26] "UNIPROT"
```

Run Gene Set Enrichment Analysis (GSEA) Biological Process

```r
gse <- gseGO(geneList=gene_list,
             ont ="BP",
             keyType = "UNIPROT",
             nPerm = 10000,
             minGSSize = 10,
             maxGSSize = 800,
```

```
            pvalueCutoff = 1,
            verbose = TRUE,
            OrgDb = organism,
            pAdjustMethod = "none")
```

preparing geneSet collections...

GSEA analysis...

Warning in .GSEA(geneList = geneList, exponent = exponent, minGSSize =
minGSSize, : We do not recommend using nPerm parameter incurrent and future
releases

Warning in fgsea(pathways = geneSets, stats = geneList, nperm = nPerm, minSize
= minGSSize, : You are trying to run fgseaSimple. It is recommended to use
fgseaMultilevel. To run fgseaMultilevel, you need to remove the nperm argument
in the fgsea function call.

Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize, gseaParam, : There are ties in
The order of those tied genes will be arbitrary, which may produce unexpected results.

## leading edge analysis...

done...

Plotting Enrichment Results

```
require(DOSE)
```

Loading required package: DOSE

DOSE v3.24.2 For help: https://yulab-smu.top/biomedical-knowledge-mining-book/
##
If you use DOSE in published research, please cite:
Guangchuang Yu, Li-Gen Wang, Guang-Rong Yan, Qing-Yu He. DOSE: an R/Bioconductor package for Disease

```
dotplot(gse,
  x = "GeneRatio",
  color = "pvalue",
  showCategory = 30,
  size = NULL,
  split = NULL,
  font.size = 12,
  title = "",
  orderBy = "x",
  label_format = 45,
  decreasing = TRUE
)
```
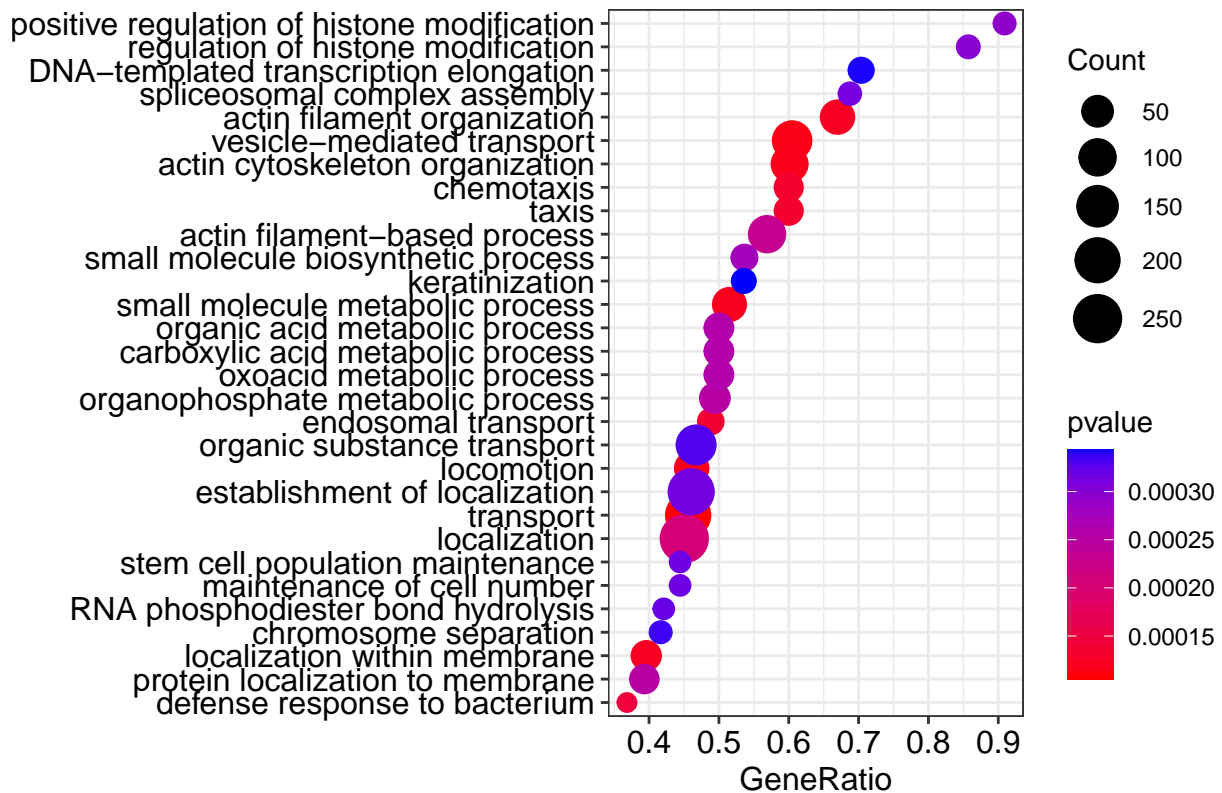
Save Plot to PDF

```r
pdf("EnrichmentTurboID.pdf", width=10, height=8)

require(DOSE)
dotplot(gse,
  x = "GeneRatio",
  color = "p.adjust",
  showCategory = 30,
  size = NULL,
  split = NULL,
  font.size = 12,
  title = "",
  label_format = 70,
)
```

Save Results to CSV

```r
resultsgseGO<-data.frame(gse@result)
write.csv(resultsgseGO, "resultsgeGOfinalTurbo.csv")
```