Package 'genekitr'

October 9, 2026

Type Package

Title Gene Analysis Toolkit

Version 1.3.0

Maintainer Yunze Liu <jieandze1314@gmail.com>

URL https://www.genekitr.fun/

BugReports https://github.com/GangLiLab/genekitr/issues

Description Provides features for searching, converting, analyzing, plotting, and exporting data effortlessly by inputting feature IDs. Enables easy retrieval of feature information, conversion of ID types, gene enrichment analysis, publication-level figures, group interaction plotting, and result export in one Excel file for seamless sharing and communication.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0)

Imports clusterProfiler, dplyr (>= 1.1.0), europepmc, fst, geneset, ggplot2 (>= 3.5.0), ggraph, ggvenn, igraph, magrittr, openxlsx, stringr, stringi, tidyr, rlang

Suggests AnnotationDbi, cowplot, ComplexUpset, forcats, fgsea, ggplotify, ggsci, ggrepel, ggridges, ggnewscale, GOplot, GOSemSim, labeling, pheatmap, tm, treemap, RColorBrewer, reshape2, rio, rrvgo, wordcloud

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Yunze Liu [aut, cre]

Repository CRAN

Date/Publication 2026-10-09 05:30:02 UTC

Contents

as.enrichdat . . . 2
Datasets . . . 3
expoSheet . . . 3
genGSEA . . . 4
genInfo . . . 5
genORA . . . 6
getPubmed . . . 7
importCP . . . 8
importPanther . . . 8
importShinygo . . . 9
plotEnrich . . . 9
plotEnrichAdv . . . 11
plotGSEA . . . 12
plotVenn . . . 14
plotVolcano . . . 15
plot_theme . . . 17
simGO . . . 18
transId . . . 18
transProbe . . . 19

Index **21**

`as.enrichdat` *Modify dataframe for enrichment plot*

Description

To make sure colname contains Description, Count, FoldEnrich/GeneRatio, pvalue/qvalue/p.adjust

Usage

```
as.enrichdat(enrich_df)
```

Arguments

`enrich_df`	Enrichment analysis ‘data.frame‘ result.

Value

‘data.frame‘

Datasets *Datasets geneList entrez gene list with decreasing fold change value*

Description

Datasets geneList entrez gene list with decreasing fold change value

Datasets Differential expression analysis result of GSE42872

Datasets msig_species contains msigdb species information

Datasets msig_category contains msigdb category information

Datasets biocOrg_name contains organism name of bioconductor

Datasets keggOrg_name contains organism name of KEGG https://www.genome.jp/kegg/catalog/org_list.html

Datasets ensOrg_name contains organism name of ensembl

Datasets hsapiens_probe_platform contains human probe platforms

expoSheet *Export list of data sets into different 'Excel' sheets*

Description

Export list of data sets into different 'Excel' sheets

Usage

```
expoSheet(
  data_list,
  data_name,
  filename = NULL,
  dir = tempdir(),
  overwrite = TRUE
)
```

Arguments

data_list	List of datasets.
data_name	Character of data names.
filename	A character string naming an xlsx file.
dir	A character string naming output directory.
overwrite	If TRUE, overwrite any existing file.

Value

An Excel file.

Examples

```
library(openxlsx)
expoSheet(
  data_list = list(mtcars, ToothGrowth),
  data_name = c("mtcars", "tooth"),
  filename = "test.xlsx", dir = tempfile()
)
```

---

genGSEA *Gene Set Enrichment Analysis*

Description

Gene Set Enrichment Analysis

Usage

```
genGSEA(
  genelist,
  geneset,
  padj_method = "BH",
  p_cutoff = 0.05,
  q_cutoff = 0.05,
  min_gset_size = 10,
  max_gset_size = 500,
  set_seed = FALSE
)
```

Arguments

genelist	Pre-ranked genelist with decreasing order, gene can be entrez, ensembl or symbol. If several ids are mapped to the same gene after id conversion (e.g. alias symbols), only the first (highest ranked) one is kept.
geneset	Gene set is a two-column data.frame with term id and gene id. Please use package ‘geneset‘ to select available gene set or make new one.
padj_method	One of "BH", "BY", "bonferroni","fdr","hochberg", "holm", "hommel", "none"
p_cutoff	Numeric of cutoff for both unadjusted and adjusted pvalue, default is 0.05.
q_cutoff	Numeric of cutoff for qvalue, default is 0.05.
min_gset_size	Numeric of minimal size of each geneset for analyzing, default is 10.
max_gset_size	Numeric of maximal size of each geneset for analyzing, default is 500.
set_seed	GSEA permutations are performed using random reordering, which causes slightly difference results after every time running. If user want to get same result every time for same input, please set ‘set_seed = TRUE‘ or ‘set.seed()‘ prior to running.

Value

A ‘data.frame‘.

Examples

```
if(requireNamespace("geneset",quietly = TRUE)){
# only gene ids
data(geneList, package = "genekitr")
gs <- geneset::getGO(org = "human",ont = "mf",data_dir = tempdir())
gse <- genGSEA(genelist = geneList, geneset = gs)
  }
```

`genInfo` *Get gene related information*

Description

Get gene related information

Usage

```
genInfo(
  id = NULL,
  org = "hs",
  unique = FALSE,
  keepNA = TRUE,
  hgVersion = c("v38", "v19")
)
```

Arguments

`id`	Gene id (symbol, ensembl or entrez id) or uniprot id. If this argument is NULL, return all gene info.
`org`	Latin organism shortname from ‘ensOrg_name‘. Default is human.
`unique`	Logical, if one-to-many mapping occurs, only keep one record with fewest NA. Default is FALSE.
`keepNA`	If some id has no match at all, keep it or not. Default is TRUE.
`hgVersion`	Select human genome build version from "v38" (default) and "v19".

Value

A ‘data.frame‘.

Examples

```
# example1: input list with fake id and one-to-many mapping id
x <- genInfo(id = c(
  "MCM10", "CDC20", "S100A9", "MMP1", "BCC7",
  "FAKEID", "TP53", "HBD", "NUDT10"
))

# example2: statistics of human gene biotypes
genInfo(org = "hs") %>%
  {
    table(.$gene_biotype)
  }

# example3: use hg19 data
x <- genInfo(id = c("TP53","BCC7"), hgVersion = "v19")

# example4: search genes with case-insensitive
x <- genInfo(id = c("tp53","nc886","FAke","EZh2"), org = "hs", unique = TRUE)
```

---

genORA *Gene Over-Representation Enrichment Analysis*

Description

Gene Over-Representation Enrichment Analysis

Usage

```
genORA(
  id,
  geneset,
  group_list = NULL,
  padj_method = "BH",
  p_cutoff = 0.05,
  q_cutoff = 0.15,
  min_gset_size = 10,
  max_gset_size = 500,
  universe
)
```

Arguments

id	A vector of gene id which can be entrezid, ensembl, symbol or uniprot.
geneset	Gene set is a two-column data.frame with term id and gene id. Please use package 'geneset' to select available gene set or make new one.
group_list	A list of gene group information, default is NULL.

padj_method	One of "BH", "BY", "bonferroni","fdr","hochberg", "holm", "hommel", "none"
p_cutoff	Numeric of cutoff for both unadjusted and adjusted pvalue, default is 0.05.
q_cutoff	Numeric of cutoff for qvalue, default is 0.15.
min_gset_size	Numeric of minimal size of each geneset for analyzing, default is 10.
max_gset_size	Numeric of maximal size of each geneset for analyzing, default is 500.
universe	Character of background genes. If missing, all genes in geneset will be used as background.

Value

A 'data.frame'.

Examples

```
# only gene ids
data(geneList, package = "genekitr")
id <- names(geneList)[abs(geneList) > 1]
gs <- geneset::getGO(org = "human",ont = "mf",data_dir = tempdir())
ora <- genORA(id, geneset = gs)

# gene id with groups
id <- c(head(names(geneList), 50), tail(names(geneList), 50))
group <- list(
  group1 = c(rep("up", 50), rep("down", 50)),
  group2 = c(rep("A", 20), rep("B", 30))
)
gora <- genORA(id, geneset = gs, group_list = group)
```

---

getPubmed *Get 'PubMed' paper records by searching abstract*

Description

PubMed<https://pubmed.ncbi.nlm.nih.gov/> is a free search engine accessing primarily the database of references and abstracts on life ciences and biomedical topics.

Usage

```
getPubmed(term, add_term = NULL, num = 100)
```

Arguments

term	query terms e.g. gene id, GO/KEGG pathway
add_term	other searching terms Default is NULL
num	limit the number of records . Default is 100.

Value

A list of ‘tibble‘ for pubmed records

Examples

```
term <- c("Tp53", "Brca1", "Tet2")
add_term <- c("stem cell", "mouse")
l <- getPubmed(term, add_term, num = 30)
# very easy to output
expoSheet(l, data_name = term, filename = "test.xlsx", dir = tempfile())
```

`importCP` *Import 'clusterProfiler' result*

Description

Import 'clusterProfiler' result

Usage

```
importCP(object, type = c("go", "gsea", "other"))
```

Arguments

`object`	clusterProfiler object.
`type`	object type from "go", "gsea" and "other". "other" includes ORA (over-representation analysis) of KEGG, DOSE,...

Value

‘data.frame‘

`importPanther` *Import 'Panther' web result*

Description

Import 'Panther' web result

Usage

```
importPanther(panther_file)
```

Arguments

panther_file Panther result file.

Value

'data.frame'

---

importShinygo *Import 'shinyGO' web result*

Description

Import 'shinyGO' web result

Usage

```
importShinygo(shinygo_file)
```

Arguments

shinygo_file ShinyGO result file.

Value

'data.frame'

---

plotEnrich *Plot for gene enrichment analysis of ORA method*

Description

Over-representation analysis (ORA) is a simple method for objectively deciding whether a set of variables of known or suspected biological relevance, such as a gene set or pathway, is more prevalent in a set of variables of interest than we expect by chance.

Usage

```
plotEnrich(
  enrich_df,
  fold_change = NULL,
 plot_type = c("bar", "wego", "dot", "bubble", "lollipop", "geneheat", "genechord",
    "network", "gomap", "goheat", "gotangram", "wordcloud", "upset"),
  term_metric = c("FoldEnrich", "GeneRatio", "Count", "RichFactor"),
  stats_metric = c("p.adjust", "pvalue", "qvalue"),
  sim_method = c("Resnik", "Lin", "Rel", "Jiang", "Wang", "JC"),
```

```
  up_color = "#E31A1C",
  down_color = "#1F78B4",
  show_gene = "all",
  xlim_left = 0,
  xlim_right = NA,
  wrap_length = NULL,
  org = NULL,
  ont = NULL,
  scale_ratio,
  layout,
  n_term,
  ...
)
```

Arguments

enrich_df	Enrichment analysis ‘data.frame‘ result.
fold_change	Fold change or logFC values with gene IDs as names. Used in "heat" and "chord" plot.
plot_type	Choose from "bar", "wego","bubble","dot", "lollipop","geneheat", "genechord", "network","gomap","goheat","gotangram","wordcloud","upset".
term_metric	Pathway term metric from one of ’GeneRatio’,’Count’,’FoldEnrich’ and ’RichFactor’.
stats_metric	Statistic metric from one of "pvalue", "p.adjust", "qvalue".
sim_method	Method of calculating the similarity between nodes, one of one of "Resnik", "Lin", "Rel", "Jiang" , "Wang" or "JC" (Jaccard’s similarity index). Only "JC" supports KEGG data. Used in "map","goheat","gotangram","wordcloud".
up_color	Color of higher statistical power (e.g. Pvalue 0.01) or higher logFC, default is "red".
down_color	Color of lower statistical power (e.g. Pvalue 1) or lower logFC, default is "blue".
show_gene	Select genes to show. Default is "all". Used in "heat" and "chord" plot.
xlim_left	X-axis left limit, default is 0.
xlim_right	X-axis right limit, default is NA.
wrap_length	Numeric, wrap text if longer than this length. Default is NULL.
org	Organism name from ‘biocOrg_name‘.
ont	One of "BP", "MF", and "CC".
scale_ratio	Numeric, scale of node and line size.
layout	Grapgh layout in "map" plot, e,g, "circle", "dh", "drl", "fr","graphopt", "grid", "lgl", "kk", "mds", "nicely" (default),"randomly", "star".
n_term	Number of terms (used in WEGO plot)
...	other arguments from ‘plot_theme‘ function

Value

A ggplot object

Examples

```
## example data
## More examples please refer to https://www.genekitr.fun/plot-ora-1.html
library(ggplot2)
data(geneList, package = "genekitr")
id <- names(geneList)[abs(geneList) > 1.5]
logfc <- geneList[id]

gs <- geneset::getGO(org = "human",ont = "bp",data_dir = tempdir())
ego <- genORA(id, geneset = gs)
ego <- ego[1:10, ]

## example plots
plotEnrich(ego, plot_type = "dot")

#plotEnrich(ego, plot_type = "bubble", scale_ratio = 0.4)

#plotEnrich(ego, plot_type = "bar")
```

---

plotEnrichAdv *Advanced Plot for gene enrichment analysis of ORA method*

Description

Over-representation analysis (ORA) is a simple method for objectively deciding whether a set of variables of known or suspected biological relevance, such as a gene set or pathway, is more prevalent in a set of variables of interest than we expect by chance.

Usage

```
plotEnrichAdv(
  up_enrich_df,
  down_enrich_df,
  plot_type = c("one", "two"),
  term_metric = c("FoldEnrich", "GeneRatio", "Count", "RichFactor"),
  stats_metric = c("p.adjust", "pvalue", "qvalue"),
  wrap_length = NULL,
  xlim_left = NULL,
  xlim_right = NULL,
  color,
  ...
)
```

Arguments

up_enrich_df	Enrichment analysis ‘data.frame‘ for up-regulated genes.
down_enrich_df	Enrichment analysis ‘data.frame‘ for down-regulated genes.
plot_type	Choose from "one" and "two". "One" represents both up and down pathways are plotted together; "two" represents up and down are plotted seperately.
term_metric	Pathway term metric from one of ’GeneRatio’,’Count’,’FoldEnrich’ and ’RichFactor’.
stats_metric	Statistic metric from one of "pvalue", "p.adjust", "qvalue".
wrap_length	Numeric, wrap text if longer than this length. Default is NULL.
xlim_left	X-axis left limit
xlim_right	X-axis right limit
color	Plot colors.
...	other arguments from ‘plot_theme‘ function

Details

Both up and down regulated pathways could be plotted in one figure as two-side barplot

Value

A ggplot object

---

plotGSEA *Plot for gene enrichment analysis of GSEA method*

Description

Gene Set Enrichment Analysis (GSEA) is a computational method that determines whether an a priori defined set of genes shows statistically significant, concordant differences between two biological states (e.g. phenotypes).

Usage

```
plotGSEA(
  gsea_list,
  plot_type = c("volcano", "classic", "fgsea", "ridge", "bar"),
  stats_metric = c("p.adjust", "pvalue", "qvalue"),
  show_pathway = NULL,
  show_gene = NULL,
  colour = NULL,
  wrap_length = NULL,
  label_by = c("id", "description"),
  ...
)
```

Arguments

gsea_list	GSEA result from ‘genGSEA‘ function
plot_type	GSEA plot type, one of ’volcano’, ’classic’, ’fgsea’, ’ridge’ or ’bar’.
stats_metric	Statistic metric from one of "pvalue", "p.adjust", "qvalue".
show_pathway	Select plotting pathways by number (will choose top N pathways) or pathway name (choose from ID column).
show_gene	Select genes to show. Default is "all". Used in "classic" plot.
colour	Colour vector. Deafault is NULL. Used in volcano, ridge and bar plot.
wrap_length	Numeric, wrap text if longer than this length. Default is NULL.
label_by	Select which column as the label. If user wants to modify labels in plot, please modify the "Description" column and set the argument as "description". Default is by ’id’.
...	other arguments transfer to ‘plot_theme‘ function

Value

A ggplot object

Examples

```
k1 = requireNamespace("cowplot",quietly = TRUE)
k2 = requireNamespace("fgsea",quietly = TRUE)
k3 = requireNamespace("ggplotify",quietly = TRUE)
k4 = requireNamespace("ggridges",quietly = TRUE)
if(k1&k2&k3&k4){
library(ggplot2)
## get GSEA result
data(geneList, package = "genekitr")
gs <- geneset::getMsigdb(org = "human",category = "H")
gse <- genGSEA(genelist = geneList, geneset = gs)

## volcano plot
# get top3 of up and down pathways
plotGSEA(gse, plot_type = "volcano", show_pathway = 3)
# choose pathway by character
pathways <- c('HALLMARK_KRAS_SIGNALING_UP','HALLMARK_P53_PATHWAY','HALLMARK_GLYCOLYSIS')
plotGSEA(gse, plot_type = "volcano", show_pathway = pathways)

## classic pathway plot
genes <- c('ENG','TP53','MET')
plotGSEA(gse, plot_type = "classic", show_pathway = pathways, show_gene = genes)

## fgsea table plot
plotGSEA(gse, plot_type = "fgsea", show_pathway = 3)

## ridgeplot
plotGSEA(gse,
  plot_type = "ridge",
```

```
  show_pathway = 10, stats_metric = "p.adjust"
)

## two-side barplot
plotGSEA(gse,
  plot_type = "bar", main_text_size = 8,
  colour = c("navyblue", "orange")
)
}
```

---

plotVenn *Venn plot for groups of genes*

Description

If gene group over 4, plot will be visulized using UpSet plot.

Usage

```
plotVenn(
  venn_list,
  use_venn = TRUE,
  color = NULL,
  alpha_degree = 0.3,
  venn_percent = FALSE,
  ...
)
```

Arguments

venn_list	A list of gene id.
use_venn	Logical, use venn to plot, default is 'TRUE', the other option is upsetplot for large list.
color	Colors for gene lists, default is NULL.
alpha_degree	Alpha transparency of each circle's area, default is 0.3.
venn_percent	Logical to show both number and percentage in venn plot.
...	other arguments transfer to 'plot_theme' function

Value

A ggplot object

Examples

```
k1 = requireNamespace("ComplexUpset",quietly = TRUE)
k2 = requireNamespace("ggsci",quietly = TRUE)
k3 = requireNamespace("RColorBrewer",quietly = TRUE)
if(k1&k2&k3){
library(ggplot2)
set1 <- paste0(rep("gene", 30), sample(1:1000, 30))
set2 <- paste0(rep("gene", 40), sample(1:1000, 40))
set3 <- paste0(rep("gene", 50), sample(1:1000, 50))
set4 <- paste0(rep("gene", 60), sample(1:1000, 60))
set5 <- paste0(rep("gene", 70), sample(1:1000, 70))
sm_gene_list <- list(gset1 = set1, gset2 = set2, gset3 = set3)
la_gene_list <- list(
  gset1 = set1, gset2 = set2, gset3 = set3,
  gset4 = set4, gset5 = set5
)
plotVenn(sm_gene_list,
  use_venn = TRUE,
  alpha_degree = 0.5,
  main_text_size = 3,
  border_thick = 0,
  venn_percent = TRUE
)
plotVenn(la_gene_list,
  use_venn = FALSE,
  main_text_size = 15,
  legend_text_size = 8,
  legend_position = 'left'
)
}
```

---

plotVolcano *Volcano plot for differential expression analysis*

Description

Volcano plot for differential expression analysis

Usage

```
plotVolcano(
  deg_df,
  stat_metric = c("p.adjust", "pvalue"),
  stat_cutoff = 0.05,
  logFC_cutoff = 1,
  up_color = "#E31A1C",
  down_color = "#1F78B4",
  other_color = "black",
```

```
  show_gene = NULL,
  dot_size = 1.75,
  alpha = 0.4,
  ...
)
```

Arguments

deg_df	DEG dataframe with gene id, logFC and stat(e.g. pvalue/qvalue).
stat_metric	Statistic metric from "pvalue" or "p.adjust".
stat_cutoff	Statistic cutoff, default is 0.05.
logFC_cutoff	Log2 fold change cutoff, default is 1 which is actually 2 fold change.
up_color	Color of up-regulated genes, default is "dark red".
down_color	Color of down-regulated genes, default is "dark blue".
other_color	Color of other genes, default is "black".
show_gene	Select genes to show, default is no genes to show.
dot_size	Volcano dot size, default is 1.75.
alpha	Volcano alpha degree, default is 0.4
...	other arguments from ‘plot_theme‘ function

Value

A ggplot object

Examples

```
if(requireNamespace("ggrepel",quietly = T)){
library(ggplot2)
data(deg, package = "genekitr")
plotVolcano(deg, "p.adjust", remove_legend = TRUE, dot_size = 3)

# show some genes
plotVolcano(deg, "p.adjust",
  remove_legend = TRUE,
  show_gene = c("CD36", "DUSP6", "IER3","CDH7")
)
}
```

plot_theme *Themes for all plots*

Description

Change ggplot text, font, legend and border

Usage

```
plot_theme(
  main_text_size = 8,
  legend_text_size = 6,
  font_type = "sans",
  border_thick = 1.5,
  remove_grid = TRUE,
  remove_border = FALSE,
  remove_main_text = FALSE,
  remove_legend_text = FALSE,
  remove_legend = FALSE
)
```

Arguments

main_text_size	Numeric, main text size
legend_text_size	Numeric, legend text size
font_type	Character, specify the plot text font family, default is "sans".
border_thick	Numeric, border thickness, default is 1. If set 0, remove both border and ticks.
remove_grid	Logical, remove background grid lines, default is FALSE.
remove_border	Logical, remove border line, default is FALSE.
remove_main_text	Logical, remove all axis text, default is FALSE.
remove_legend_text	Logical, remove all legend text, default is FALSE.
remove_legend	Logical, remove entire legend, default is FALSE.

Value

ggplot theme

Examples

```
library(ggplot2)
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  plot_theme(font_type = "Times", border_thick = 2)
```

simGO *Simplify GO enrichment result*

Description

The Gene Ontology (GO) is a major bioinformatics initiative to unify the representation of gene and gene product attributes across all species.

Usage

```
simGO(
  enrich_df,
  sim_method = c("Resnik", "Lin", "Rel", "Jiang", "Wang"),
  org = NULL,
  ont = NULL
)
```

Arguments

enrich_df	GO enrichment analysis of ‘genORA()‘ result.
sim_method	Method of calculating the similarity between nodes, one of one of "Resnik", "Lin", "Rel", "Jiang" , "Wang" methods.
org	Organism name from ‘biocOrg_name‘.
ont	One of "bp", "mf", and "cc".

Value

A ‘data.frame‘ contains simplified GO terms.

transId *Transform id among symbol, entrezid, ensembl and uniprot.*

Description

Transform id among symbol, entrezid, ensembl and uniprot.

Usage

```
transId(
  id,
  transTo,
  org = "hs",
  unique = FALSE,
  keepNA = FALSE,
  hgVersion = c("v38", "v19")
)
```

Arguments

id	Gene ids or protein ids.
transTo	Transform to what type. User could select one or more from "symbol", "entrez", "ensembl" or "uniprot."
org	Latin organism shortname from ‘ensOrg_name‘. Default is human.
unique	Logical, if one-to-many mapping occurs, only keep one record with fewest NA. Default is FALSE.
keepNA	If some id has no match at all, keep it or not. Default is FALSE.
hgVersion	Select human genome build version from "v38" (default) and "v19".

Value

data frame, first column is input id and others are converted id.

Examples

```
# example1:
transId(
  id = c("Cyp2c23", "Fhit", "Gal3st2b", "Trp53", "Tp53"),
  transTo = "ensembl", org = "mouse", keepNA = FALSE
)

## example2: input id with one-to-many mapping and fake one
transId(
  id = c("MMD2", "HBD", "RNR1", "TEC", "BCC7", "FAKEID", "TP53"),
  transTo = c("entrez", "ensembl"), keepNA = TRUE
)

# example3: auto-recognize ensembl version number
transId("ENSG00000141510.11", "symbol")

# example4: search genes with case-insensitive
transId(c('nc886','ezh2','TP53'),transTo = "ensembl",org = 'hs',unique = TRUE)
```

---

transProbe *Transform probe id to symbol, entrezid, ensembl or uniprot.*

Description

Transform probe id to symbol, entrezid, ensembl or uniprot.

Usage

```
transProbe(id, transTo, org = "human", platform = NULL)
```

Arguments

id	probe ids.
transTo	Transform to what type. User could select one or more from "symbol", "entrez", "ensembl" or "uniprot."
org	'human'.
platform	Probe platform. If NULL, program will detect automatically.

Value

data frame, first column is probe id and others are converted id.

Index

∗ **datasets**
    Datasets, 3

as.enrichdat, 2

biocOrg_name *(*Datasets*)*, 3

Datasets, 3
deg *(*Datasets*)*, 3

ensOrg_name *(*Datasets*)*, 3
expoSheet, 3

geneList *(*Datasets*)*, 3
genGSEA, 4
genInfo, 5
genORA, 6
getPubmed, 7

hsapiens_probe_platform *(*Datasets*)*, 3

importCP, 8
importPanther, 8
importShinygo, 9

keggOrg_name *(*Datasets*)*, 3

msig_category *(*Datasets*)*, 3
msig_org *(*Datasets*)*, 3

plot_theme, 17
plotEnrich, 9
plotEnrichAdv, 11
plotGSEA, 12
plotVenn, 14
plotVolcano, 15

simGO, 18

transId, 18
transProbe, 19