

# Package ‘genekitr’

January 20, 2023

**Type** Package

**Title** Gene Analysis Toolkit

**Version** 1.1.0

**Maintainer** Yunze Liu <jieandze1314@gmail.com>

**URL** <https://www.genekitr.fun/>

**BugReports** <https://github.com/GangLiLab/genekitr/issues>

**Description** An analysis toolkit focused on genes. It mainly includes five features (search, convert, analysis, plot, and export).

The user just needs to input feature id ('entrez', 'symbol', 'ensembl' or 'uniprot') to retrieve feature information and 'PubMed' <<https://pubmed.ncbi.nlm.nih.gov/>> records,

to convert id types, to easily handle gene enrichment analysis and publication-level figures, to plot group interaction and export results as sheets in one 'excel' file to easily share and communicate with others.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.6)

**Imports** clusterProfiler, dplyr, europepmc, fst, geneset, ggplot2, ggraph, igraph, magrittr, openxlsx, stringr, stringi, tidy, VennDiagram, rlang

**Suggests** AnnotationDbi, BiocManager, cowplot, ComplexUpset, data.table, forcats, fgsea, futile.logger, ggplotify, ggsci, ggupset, ggrepel, ggridges, ggnewscale, GOplot, GOSemSim, labeling, msigdb, pheatmap, tm, treemap, RColorBrewer, rappdirs, RCurl, reshape2, rio, rrvgo, scales, stats, testthat (>= 3.0.0), tibble, wordcloud, knitr, rmarkdown

**RoxygenNote** 7.2.2

**NeedsCompilation** no

**Author** Yunze Liu [aut, cre] (<<https://orcid.org/0000-0002-7414-8556>>)

**Repository** CRAN

**Date/Publication** 2023-01-20 03:30:02 UTC

## R topics documented:

as.enrichdat . . . . .	2
Datasets . . . . .	3
expoSheet . . . . .	3
genGSEA . . . . .	4
genInfo . . . . .	5
genORA . . . . .	6
getPubmed . . . . .	7
importCP . . . . .	8
importPanther . . . . .	8
plotEnrich . . . . .	9
plotEnrichAdv . . . . .	11
plotGSEA . . . . .	12
plotVenn . . . . .	14
plotVolcano . . . . .	15
plot_theme . . . . .	16
simGO . . . . .	17
transId . . . . .	18
transProbe . . . . .	19
<b>Index</b>	<b>20</b>

---

as.enrichdat	<i>Modify dataframe for enrichment plot</i>
--------------	---

---

### 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	<i>Datasets geneList entrez gene list with decreasing fold change value</i>
----------	---

---

**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](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	<i>Export list of data sets into different 'Excel' sheets</i>
-----------	---

---

**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.
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**

```
# only gene ids
data(geneList, package = "genekitr")
gs <- geneset::getGO(org = "human", ont = "mf")
gse <- genGSEA(genelist = geneList, geneset = gs)
```

---

genInfo	<i>Get gene related information</i>
---------	-------------------------------------

---

**Description**

Get gene related information

**Usage**

```
genInfo(id = NULL, org = "hs", unique = FALSE, keepNA = TRUE)
```

**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.

**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)
}
```

---

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")
ora <- genORA(id, geneset = gs)

# gene id with groups
id <- c(head(names(geneList), 100), tail(names(geneList), 100))
group <- list(
  group1 = c(rep("up", 100), rep("down", 100)),
  group2 = c(rep("A", 130), rep("B", 70))
)
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 sciences 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	<i>Import 'clusterProfiler' result</i>
----------	--

---

**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	<i>Import 'Panther' web result</i>
---------------	------------------------------------

---

**Description**

Import 'Panther' web result

**Usage**

```
importPanther(panther_file)
```

**Arguments**

panther_file	Panther 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	Graph 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
library(ggplot2)
library(igraph)
library(ggraph)
data(geneList, package = "genekitr")
id <- names(geneList)[abs(geneList) > 1.5]
logfc <- geneList[id]

gs <- geneset::getGO(org = "human", ont = "bp")
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")

plotEnrich(ego,
  plot_type = "lollipop",
  down_color = "#325CAC", up_color = "#E69056",
  wrap_length = 25, scale_ratio = 0.4
)

plotEnrich(ego, plot_type = "geneheat")
```

```

show_gene <- c("BRCA2", "CDK1", "JUN", "MCM8", "TIPIN")
plotEnrich(ego, plot_type = "geneheat", show_gene = show_gene)
plotEnrich(ego, fold_change = logfc, plot_type = "geneheat", show_gene = show_gene)

plotEnrich(ego, fold_change = logfc, plot_type = "genechord", show_gene = show_gene)

plotEnrich(ego, plot_type = "network", scale_ratio = 0.5)

plotEnrich(ego, plot_type = "gomap", wrap_length = 25)

plotEnrich(ego, plot_type = "goheat", sim_method = "Rel")

plotEnrich(ego, plot_type = "gotangram", sim_method = "Rel")

plotEnrich(ego, plot_type = "wordcloud")

plotEnrich(ego, plot_type = "upset", main_text_size = 15, legend_text_size = 8)

```

---

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 separately.
term_metric	Pathway term metric from one of 'GeneRatio', 'Count', 'FoldEnrich' and 'Rich-Factor'.
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,
  ...
)
```

**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.
...	other arguments transfer to 'plot_theme' function

**Value**

A ggplot object

**Examples**

```

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 visualized using UpSet plot.

### Usage

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

### 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.
...	other arguments transfer to 'plot_theme' function

### Value

A ggplot object

### Examples

```
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,
  color = ggsci::pal_lancet()(3),
  alpha_degree = 1,
  main_text_size = 1.5,
  border_thick = 0
)
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",  
  show_gene = NULL,  
  dot_size = 1.75,  
  ...  
)
```

## 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".
show_gene	Select genes to show, default is no genes to show.
dot_size	Volcano dot size, default is 1.75.
...	other arguments from 'plot_theme' function

## Value

A ggplot object

**Examples**

```

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	<i>Transform id among symbol, entrezid, ensembl and uniprot.</i>
---------	--

---

**Description**

Transform id among symbol, entrezid, ensembl and uniprot.

**Usage**

```
transId(id, transTo, org = "hs", unique = FALSE, keepNA = FALSE)
```

**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.

**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")
```

---

transProbe	<i>Transform probe id to symbol, entrezid, ensembl or uniprot.</i>
------------	--

---

**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.

**Examples**

```
## Not run:  
data(deg, package = "genekitr")  
id <- deg$probe_id  
transProbe(id, transTo = "symbol")  
  
## End(Not run)
```

# 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](#)

keggOrg\_name (Datasets), [3](#)

msig\_category (Datasets), [3](#)

msig\_org (Datasets), [3](#)

plot\_theme, [16](#)

plotEnrich, [9](#)

plotEnrichAdv, [11](#)

plotGSEA, [12](#)

plotVenn, [14](#)

plotVolcano, [15](#)

simGO, [17](#)

transId, [18](#)

transProbe, [19](#)