Package 'BioEnricher'

December 5, 2023

Type Package

Title Integrate analysis and visualization for bioinformatic enrichment analyzer

Version 0.1.0

Author Zaoqu Liu

Description BioEnricher lies in addressing two issues: firstly, it facilitates the seamless integration for enrichment analysis, encompassing diverse functionalities such as GO, KEGG, WikiPathways, Reactome, MsigDB, Disease Ontology, Cancer Gene Network, DisGeNET, CellMarker, and CMAP (drugs); infers the activities of transcription factors and PROGENy cancer pathways; searches the gene information, PubMed records and GEO metadata based on the input terms. Secondly, it encapsulates advanced visualization functions, streamlining the process for faster and more convenient data presentation.

License MIT + file LICENSE

Encoding UTF-8

ByteCompile true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Depends R (>= 4.3.0)

Imports broom,
clusterProfiler,
dorothea,
DOSE,
dplyr,
enrichplot,
europepmc,
ggplot2,
GSVA,
HGNChelper,
Hmisc,
httr,
jsonlite,
magrittr,
msigdbr,
openssl,
pathview,
png,
progeny,

purrr,
ReactomePA,
rlang,
stats,
stringr,
viper,
vroom

Suggests knitr, rmarkdown

VignetteBuilder knitr

biocViews Software, GeneSetEnrichment, Visualization

BugReports https://github.com/Zaoqu-Liu/BioEnricher/issues

URL https://github.com/Zaoqu-Liu/BioEnricher

R topics documented:

CMAPfromDSEATM *A dataframe including drugs and their related genes*

Description

A dataframe including drugs and their related genes

Usage

```
CMAPfromDSEATM
```

Format

A dataframe with four columns from DSEATM

cols_brown_green *A vector of colors*

Description

A vector of colors

Usage

```
cols_brown_green
```

Format

A vector with 11 types of colors.

gene.info *Get gene related information*

Description

Get the basic information of genes. This function is from genekitr package.

Usage

```
gene.info(
  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

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

# use hg19 data
x <- gene.info(id = c("TP53", "BCC7"), hgVersion = "v19")
```

---

listEnrichMethod *List of enrichment methods*

Description

List of enrichment methods, including GO, KEGG, MKEGG, WikiPathways, Reactome, MsigDB, DO, CGN, DisGeNET, CellMarker, and CMAP.

Usage

```
listEnrichMethod()
```

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
listEnrichMethod()
```

lzq_getEF *Get enrichment factor from enrichResult*

Description

Get enrichment factor from enrichResult (clusterProfiler).

Usage

```
lzq_getEF(res)
```

Arguments

res	enrichResult from clusterProfiler.

Value

A new result with enrichment factor.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
genes <- c("CANX", "HSPA1B", "KLRC2", "PSMC6", "RFXAP", "TAP1")
obj <- clusterProfiler::enrichGO(genes, org.Hs.eg.db::org.Hs.eg.db,
  keyType = "SYMBOL", ont = "BP"
)
obj2 <- BioEnricher::lzq_getEF(obj)
obj2@result$EnrichmentFactor
```

lzq_getGR_BR *Get numeric GeneRatio and BgRatio from enrichResult*

Description

Get numeric GeneRatio and BgRatio from enrichResult (clusterProfiler).

Usage

```
lzq_getGR_BR(res)
```

Arguments

res	enrichResult from clusterProfiler.

Value

A new result with numeric GeneRatio and BgRatio.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
genes <- c("CANX", "HSPA1B", "KLRC2", "PSMC6", "RFXAP", "TAP1")
obj <- clusterProfiler::enrichGO(genes, org.Hs.eg.db::org.Hs.eg.db,
  keyType = "SYMBOL", ont = "BP"
)
obj2 <- BioEnricher::lzq_getGR_BR(obj)
obj2@result$GeneRatio
obj2@result$BgRatio
```

---

lzq_GSEA *Gene set enrichment analysis*

Description

Perform gene set enrichment analysis included GO, KEGG, WikiPathways, Reactome, MsigDB, Disease Ontoloty, Cancer Gene Network, DisGeNET, CellMarker, and CMAP.

Usage

```
lzq_GSEA(
  genes,
  gene.type = "SYMBOL",
  enrich.type,
  organism = "Human",
  GO.ont = "BP",
  GO.simplify = T,
  KEGG.use.internal.data = F,
  MsigDB.category = "H",
  CMAP.min.Geneset.Size = 3,
  pvalue.cutoff = 0.05,
  padjust.method = "BH",
  min.Geneset.Size = 10,
  max.Geneset.Size = 1000
)
```

Arguments

genes	An order ranked geneList.
gene.type	Keytype of input gene.
enrich.type	Select an enrichment method. One of GO, KEGG, MKEGG, WikiPathways, Reactome, MsigDB, DO, CGN, DisGeNET, CellMarker, and CMAP. WikiPathways can be replaced by WP, Reactome can be replaced by RP, and CellMarker can be replaced by CM.
organism	Specify species, currently support only Human and Mouse.
GO.ont	GO parameter. One of "BP", "MF", and "CC" subontologies, or "ALL" for all three.

`GO.simplify` GO parameter. Whether to remove redundancy of enriched GO terms.

`KEGG.use.internal.data`
KEGG parameter. Logical, use KEGG.db or latest online KEGG data.

`MsigDB.category`
MsigDB parameter. MSigDB collection abbreviation, such as All, H, C1, C2, C3, C4, C5, C6, C7.

`CMAP.min.Geneset.Size`
CMAP parameter. Minimal size of CMAP genes annotated for testing. Recommended use 3.

`pvalue.cutoff` pvalue cutoff on enrichment tests to report as significant.

`padjust.method` one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

`min.Geneset.Size`
Minimal size of genes annotated for testing. Not suitable for CMAP.

`max.Geneset.Size`
Maximal size of genes annotated for testing.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
library(airway)
library(DESeq2)
library(tidyverse)
library(clusterProfiler)
library(org.Hs.eg.db)
data(airway)
se <- airway
se$dex <- relevel(se$dex, "untrt")
res <- DESeqDataSet(se, design = ~ cell + dex) %>%
  estimateSizeFactors() %>%
  DESeq() %>%
  results() %>%
  as.data.frame() %>%
  na.omit()
ann <- bitr(rownames(res), "ENSEMBL", "SYMBOL", org.Hs.eg.db)
res <- merge(ann, res, by.x = 1, by.y = 0) %>% dplyr::distinct(SYMBOL, .keep_all = TRUE)

# Obtain an order ranked geneList.
grlist <- res$log2FoldChange
names(grlist) <- res$SYMBOL
grlist <- sort(grlist, decreasing = TRUE)

# Set enrich.type using an enrichment analysis method mentioned above.
fit <- lzq_GSEA(grlist, enrich.type = "GO")
```

`lzq_GSEA.barplot1` *Enrichment barplot for positive or negative GSEA results*

Description

Plot enrichment barplot for positive or negative GSEA results.

Usage

```
lzq_GSEA.barplot1(
  enrich.obj,
  type = "Positive",
  show.term.num = 15,
  Selct.P = "FDR",
  cutoff.P = 0.05,
  colors = rev(cols_brown_green),
  add.bar.border = T,
  bar.width = 0.6,
  y.label.position = "right",
  title = NULL,
  legend.position = "bottom",
  theme.plot = theme_bw(base_rect_size = 1.5),
  use.Chinese = F,
  appid = "20231122001888718",
  key = "5GpDqe8F3pmXfnOkEKGQ"
)
```

Arguments

`enrich.obj`	A GSEA enrichment object from clusterProfiler.
`type`	Specify whether you want to show positive or negative results.
`show.term.num`	A number or a list of terms. If it is a number, the first n terms will be displayed. If it is a list of terms, the selected terms will be displayed.
`Selct.P`	Nominal P value (NP) or adjust P value (FDR) were selected to define significant terms.
`cutoff.P`	A cutoff value for Select_P.
`colors`	A color vector for the bars.
`add.bar.border`	Logical. Whether to add the black border of bars.
`bar.width`	Width of bar in the plot.
`y.label.position`	Y label position. right or left.
`title`	Title of the plot.
`legend.position`	Position of legend. 'none', 'right', 'left' or two numeric variables.
`theme.plot`	ggtheme of plot.
`use.Chinese`	Logical. Whether to use Chinese annotation in the barplot.
`appid`	User app id from baidu translation api. https://fanyi-api.baidu.com/manage/developer.
`key`	User Key from baidu translation api. https://fanyi-api.baidu.com/manage/developer.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
library(airway)
library(DESeq2)
library(tidyverse)
library(clusterProfiler)
library(org.Hs.eg.db)
data(airway)
se <- airway
se$dex <- relevel(se$dex, "untrt")
res <- DESeqDataSet(se, design = ~ cell + dex) %>%
  estimateSizeFactors() %>%
  DESeq() %>%
  results() %>%
  as.data.frame() %>%
  na.omit()
ann <- bitr(rownames(res), "ENSEMBL", "SYMBOL", org.Hs.eg.db)
res <- merge(ann, res, by.x = 1, by.y = 0) %>% dplyr::distinct(SYMBOL, .keep_all = TRUE)

# Obtain an order ranked geneList.
grlist <- res$log2FoldChange
names(grlist) <- res$SYMBOL
grlist <- sort(grlist, decreasing = TRUE)

# Integrative enrichment analysis of the ranked gene list
fit <- lzq_GSEA.integrated(genes = grlist)

lzq_GSEA.barplot1(enrich.obj = fit$simplyGO, type = "pos")
```

---

lzq_GSEA.barplot2 *Enrichment barplot for positive and negative GSEA results*

Description

Plot enrichment barplot for positive and negative GSEA results.

Usage

```
lzq_GSEA.barplot2(
  enrich.obj,
  Selct.P = "FDR",
  cutoff.P = 0.05,
  types = c("Positive", "Negative"),
  type.colors = c("#ED6355", "#3E94B5"),
  pos.top.pathway.num = 10,
  neg.top.pathway.num = 10,
  bar.width = 0.6,
  add.bar.border = T,
  x.limit.fold = 1.05,
  label.size = 3.5,
```

```
  legend.position = "bottom",
  use.Chinese = F,
  appid = "20231122001888718",
  key = "5GpDqe8F3pmXfnOkEKGQ"
)
```

Arguments

enrich.obj	A GSEA enrichment object from clusterProfiler.
Selct.P	Nominal P value (NP) or adjust P value (FDR) were selected to define significant terms.
cutoff.P	A cutoff value for Select_P.
types	Two characters for defining the types of two objects.
type.colors	Two colors for the types of two objects.
pos.top.pathway.num	The number of top pathways in positive terms. Based on the significant test.
neg.top.pathway.num	The number of top pathways in negative terms. Based on the significant test.
bar.width	Width of bar in the plot.
add.bar.border	Logical. Whether to add the black border of bars.
x.limit.fold	Specify the fold of x limitation. Because some terms is too long.
label.size	Fontsize of label.
legend.position	none, left, right, top, bottom; Or Two numeric variables indicated x and y positions, respectively.
use.Chinese	Logical. Whether to use Chinese annotation in the barplot.
appid	User app id from baidu translation api. https://fanyi-api.baidu.com/manage/developer.
key	User Key from baidu translation api. https://fanyi-api.baidu.com/manage/developer.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
library(airway)
library(DESeq2)
library(tidyverse)
library(clusterProfiler)
library(org.Hs.eg.db)
data(airway)
se <- airway
se$dex <- relevel(se$dex, "untrt")
res <- DESeqDataSet(se, design = ~ cell + dex) %>%
  estimateSizeFactors() %>%
  DESeq() %>%
  results() %>%
  as.data.frame() %>%
  na.omit()
ann <- bitr(rownames(res), "ENSEMBL", "SYMBOL", org.Hs.eg.db)
res <- merge(ann, res, by.x = 1, by.y = 0) %>% dplyr::distinct(SYMBOL, .keep_all = TRUE)
```

```
# Obtain an order ranked geneList.
grlist <- res$log2FoldChange
names(grlist) <- res$SYMBOL
grlist <- sort(grlist, decreasing = TRUE)

# Integrative enrichment analysis of the ranked gene list
fit <- lzq_GSEA.integrated(genes = grlist)

lzq_GSEA.barplot2(enrich.obj = fit$simplyGO)
```

`lzq_GSEA.dotplot1` *Enrichment dotplot for positive or negative GSEA results*

Description

Plot enrichment dotplot for positive or negative GSEA results.

Usage

```
lzq_GSEA.dotplot1(
  enrich.obj,
  type = "neg",
  show.term.num = 15,
  Selct.P = "FDR",
  cutoff.P = 0.05,
  colors = rev(cols_brown_green),
  size.range = c(3, 8),
  y.label.position = "right",
  title = NULL,
  legend.position = "bottom",
  theme.plot = theme_bw(base_rect_size = 1.5),
  use.Chinese = F,
  appid = "20231122001888718",
  key = "5GpDqe8F3pmXfnOkEKGQ"
)
```

Arguments

`enrich.obj`	A GSEA enrichment object from clusterProfiler.
`type`	Specify whether you want to show positive or negative results.
`show.term.num`	A number or a list of terms. If it is a number, the first n terms will be displayed. If it is a list of terms, the selected terms will be displayed.
`Selct.P`	Nominal P value (NP) or adjust P value (FDR) were selected to define significant terms.
`cutoff.P`	A cutoff value for Select_P.
`colors`	A color vector for the bars.
`size.range`	Two numeric variables, the first is minimal value and the first is maximal value.
`y.label.position`	Y label position. right or left.

title	Title of the plot.
legend.position	Position of legend. 'none', 'right', 'left' or two numeric variables.
theme.plot	ggtheme of plot.
use.Chinese	Logical. Whether to use Chinese annotation in the barplot.
appid	User app id from baidu translation api. https://fanyi-api.baidu.com/manage/developer.
key	User Key from baidu translation api. https://fanyi-api.baidu.com/manage/developer.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
library(airway)
library(DESeq2)
library(tidyverse)
library(clusterProfiler)
library(org.Hs.eg.db)
data(airway)
se <- airway
se$dex <- relevel(se$dex, "untrt")
res <- DESeqDataSet(se, design = ~ cell + dex) %>%
  estimateSizeFactors() %>%
  DESeq() %>%
  results() %>%
  as.data.frame() %>%
  na.omit()
ann <- bitr(rownames(res), "ENSEMBL", "SYMBOL", org.Hs.eg.db)
res <- merge(ann, res, by.x = 1, by.y = 0) %>% dplyr::distinct(SYMBOL, .keep_all = TRUE)

# Obtain an order ranked geneList.
grlist <- res$log2FoldChange
names(grlist) <- res$SYMBOL
grlist <- sort(grlist, decreasing = TRUE)

# Integrative enrichment analysis of the ranked gene list
fit <- lzq_GSEA.integrated(genes = grlist)

lzq_GSEA.dotplot1(enrich.obj = fit$simplyGO, type = "pos")
```

---

lzq_GSEA.integrated *Integrate gene set enrichment analysis*

Description

Perform integrated gene set enrichment analysis included GO, KEGG, WikiPathways, Reactome, MsigDB, Disease Ontoloty, Cancer Gene Network, DisGeNET, CellMarker, and CMAP.

Usage

```
lzq_GSEA.integrated(
  genes,
  gene.type = "SYMBOL",
  organism = "Human",
  GO.ont = "BP",
  KEGG.use.internal.data = F,
  perform.WikiPathways = F,
  perform.Reactome = F,
  perform.MsigDB = F,
  MsigDB.category = "H",
  perform.disease.ontoloty = F,
  perform.Cancer.Gene.Network = F,
  perform.DisGeNET = F,
  perform.CellMarker = F,
  perform.CMAP = T,
  pvalue.cutoff = 0.05,
  padjust.method = "BH",
  min.Geneset.Size = 10,
  max.Geneset.Size = 1000,
  CMAP.min.Geneset.Size = 3
)
```

Arguments

genes	A vector of gene id.
gene.type	Keytype of input gene.
organism	Specify species, currently support only Human and Mouse.
GO.ont	One of "BP", "MF", and "CC" subontologies, or "ALL" for all three.
KEGG.use.internal.data	Logical, use KEGG.db or latest online KEGG data.
perform.WikiPathways	Whether to perform WikiPathways enrichment.
perform.Reactome	Whether to perform Reactome enrichment.
perform.MsigDB	Whether to perform MsigDB enrichment.
MsigDB.category	MSigDB collection abbreviation, such as All, H, C1, C2, C3, C4, C5, C6, C7.
perform.disease.ontoloty	Whether to perform DO enrichment.
perform.Cancer.Gene.Network	Whether to perform CGN enrichment.
perform.DisGeNET	Whether to perform DisGeNET enrichment.
perform.CellMarker	Whether to perform CellMarker enrichment. Marker from cellmarker database.
perform.CMAP	Whether to perform CMAP enrichment. Marker from CMAP database (in DSEATM tool).
pvalue.cutoff	pvalue cutoff on enrichment tests to report as significant.

padjust.method one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

min.Geneset.Size
: Minimal size of genes annotated for testing. Not suitable for CMAP.

max.Geneset.Size
: Maximal size of genes annotated for testing.

CMAP.min.Geneset.Size
: Minimal size of CMAP genes annotated for testing. Recommended use 3.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
library(airway)
library(DESeq2)
library(tidyverse)
library(clusterProfiler)
library(org.Hs.eg.db)
data(airway)
se <- airway
se$dex <- relevel(se$dex, "untrt")
res <- DESeqDataSet(se, design = ~ cell + dex) %>%
  estimateSizeFactors() %>%
  DESeq() %>%
  results() %>%
  as.data.frame() %>%
  na.omit()
ann <- bitr(rownames(res), "ENSEMBL", "SYMBOL", org.Hs.eg.db)
res <- merge(ann, res, by.x = 1, by.y = 0) %>% dplyr::distinct(SYMBOL, .keep_all = TRUE)

# Obtain an order ranked geneList.
grlist <- res$log2FoldChange
names(grlist) <- res$SYMBOL
grlist <- sort(grlist, decreasing = TRUE)

# Integrative enrichment analysis of the ranked gene list
fit <- lzq_GSEA.integrated(genes = grlist)
```

---

lzq_gseaplot *Visualize analyzing result of GSEA.*

Description

Visualize analyzing result of GSEA.

Usage

```
lzq_gseaplot(
  GSEA.result,
  Pathway.ID,
  heatbar = T,
  rank = T,
```

```
  line.color = "#41A98E",
  rank.colors = viridis::viridis(10),
  heatbar.colors = c(rev(RColorBrewer::brewer.pal(5, "Blues")),
    RColorBrewer::brewer.pal(5, "Reds")),
  add.x.ann = T,
  x.lab = "Gene ranks",
  line.y.lab = "Enrichment score",
  rank.y.lab = "logFC",
  statistic.position = c(0.5, 0.2),
  statistic.face = "italic",
  statistic.size = 3.5,
  rel.heights = c(1.5, 0.2, 1),
  theme.plot = theme_bw(base_rect_size = 1.5)
)
```

Arguments

GSEA.result	GSEA results from clusterProfiler::GSEA() function.
Pathway.ID	Corresponding pathway term of the output plot.
heatbar	Whether to add heatbar. Default True.
rank	Whether to add Rank map. Default True.
line.color	Line color for running score.
rank.colors	Color scheme of rank lines. A vector.
heatbar.colors	Color scheme of heatbar. A vector.
add.x.ann	Whether to add the title, text, and ticks of X axis.
x.lab	X label.
line.y.lab	Y label of running score plot.
rank.y.lab	Y label of rank plot.
statistic.position	Position of statistics in the running score plot.
statistic.face	Font face of statistics.
statistic.size	Font size of statistics.
rel.heights	Relative heights of subplots.
theme.plot	A theme object from ggplot2.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
library(airway)
library(DESeq2)
library(tidyverse)
library(clusterProfiler)
library(org.Hs.eg.db)
data(airway)
se <- airway
se$dex <- relevel(se$dex, "untrt")
```

```
res <- DESeqDataSet(se, design = ~ cell + dex) %>%
  estimateSizeFactors() %>%
  DESeq() %>%
  results() %>%
  as.data.frame() %>%
  na.omit()
ann <- bitr(rownames(res), "ENSEMBL", "SYMBOL", org.Hs.eg.db)
res <- merge(ann, res, by.x = 1, by.y = 0) %>% dplyr::distinct(SYMBOL, .keep_all = TRUE)

# Obtain an order ranked geneList.
grlist <- res$log2FoldChange
names(grlist) <- res$SYMBOL
grlist <- sort(grlist, decreasing = TRUE)

# Integrative enrichment analysis of the ranked gene list
fit <- lzq_GSEA.integrated(genes = grlist)
lzq_gseaplot(
  fit$simplyGO,
  Pathway.ID = "GO:0030016",
  rank = F,
  statistic.position = c(0.71, 0.85),
  rel.heights = c(1, 0.4)
)
```

`lzq_inferTF` *Perform VIPER analysis*

Description

This function performs Virtual Inference of Protein-activity by Enriched Regulon analysis

Usage

```
lzq_inferTF(
  exp,
  organism = "Human",
  use.cancer.regulons = F,
  confidence = c("A", "B", "C")
)
```

Arguments

`exp`	Numeric matrix containing the expression data or gene expression signatures, with samples in columns and genes in rows.
`organism`	Specify species, currently support only Human and Mouse.
`use.cancer.regulons`	Use TF-target interactions for cancer application.
`confidence`	The score comprises five categories, ranging from A (highest confidence) to E (lowest confidence). The scoring criteria are described in PMID: 31340985.

Value

A matrix of inferred activity for each regulator gene in the network across all samples.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
gene_expression <- as.matrix(read.csv(system.file("extdata", "human_input.csv", package = "progeny"),
  row.names = 1
))
s <- lzq_inferTF(gene_expression)
```

---

lzq_KEGGview *KEGG pathway visualization*

Description

Simple visualization of KEGG pathway based on pathview package.

Usage

```
lzq_KEGGview(
  gene.data = NULL,
  gene.type = "SYMBOL",
  pathway.id,
  species = "hsa",
  figure.suffix = ""
)
```

Arguments

gene.data	either vector (single sample) or a matrix-like data (multiple sample). Vector should be numeric with gene IDs as names or it may also be character of gene IDs. Character vector is treated as discrete or count data. Matrix-like data structure has genes as rows and samples as columns. Row names should be gene IDs. Here gene ID is a generic concepts, including multiple types of gene, transcript and protein uniquely mappable to KEGG gene IDs. KEGG ortholog IDs are also treated as gene IDs as to handle metagenomic data. Check details for mappable ID types. Default gene.data=NULL.
gene.type	character, ID type used for the gene.data, case insensitive. Default gene.idtype="entrez", i.e. Entrez Gene, which are the primary KEGG gene ID for many common model organisms. For other species, gene.idtype should be set to "KEGG" as KEGG use other types of gene IDs. For the common model organisms (to check the list, do: data(bods); bods), you may also specify other types of valid IDs. To check the ID list, do: data(gene.idtype.list); gene.idtype.list.
pathway.id	character vector, the KEGG pathway ID(s), usually 5 digit, may also include the 3 letter KEGG species code.
species	character, either the kegg code, scientific name or the common name of the target species. This applies to both pathway and gene.data or cpd.data. When KEGG ortholog pathway is considered, species="ko". Default species="hsa", it is equivalent to use either "Homo sapiens" (scientific name) or "human" (common name).

figure.suffix character, the suffix to be added after the pathway name as part of the output graph file. Sample names or column names of the gene.data or cpd.data are also added when there are multiple samples. Default out.suffix="pathview".

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
library(airway)
library(DESeq2)
library(tidyverse)
library(clusterProfiler)
library(org.Hs.eg.db)
data(airway)
se <- airway
se$dex <- relevel(se$dex, "untrt")
res <- DESeqDataSet(se, design = ~ cell + dex) %>%
  estimateSizeFactors() %>%
  DESeq() %>%
  results() %>%
  as.data.frame() %>%
  na.omit()
ann <- bitr(rownames(res), "ENSEMBL", "SYMBOL", org.Hs.eg.db)
res <- merge(ann, res, by.x = 1, by.y = 0) %>% dplyr::distinct(SYMBOL, .keep_all = TRUE)

# Set enrich.type using an enrichment analysis method mentioned above.
kegg <- lzq_ORA(
  genes = res$SYMBOL[res$log2FoldChange > 0 & res$padj < 0.05],
  enrich.type = "KEGG"
)

res2 <- res[res$log2FoldChange > 0 & res$padj < 0.05, c(2, 4)]
res2 <- data.frame(row.names = res2$SYMBOL, R = res2$log2FoldChange)

lzq_KEGGview(gene.data = res2, pathway.id = "hsa04218")
```

---

lzq_ORA *Over-representative analysis*

Description

Perform over-representative analysis included GO, KEGG, WikiPathways, Reactome, MsigDB, Disease Ontoloty, Cancer Gene Network, DisGeNET, CellMarker, and CMAP.

Usage

```
lzq_ORA(
  genes,
  background.genes = NULL,
  gene.type = "SYMBOL",
  enrich.type,
  organism = "Human",
```

```
  GO.ont = "BP",
  GO.simplify = T,
  KEGG.use.internal.data = F,
  MsigDB.category = "H",
  CMAP.min.Geneset.Size = 3,
  pvalue.cutoff = 0.05,
  qvalue.cutoff = 0.05,
  padjust.method = "BH",
  min.Geneset.Size = 10,
  max.Geneset.Size = 1000
)
```

Arguments

genes	A vector of gene id.
background.genes	Background genes. If missing, the all genes listed in the database (eg TERM2GENE table) will be used as background.
gene.type	Keytype of input gene.
enrich.type	Select an enrichment method. One of GO, KEGG, MKEGG, WikiPathways, Reactome, MsigDB, DO, CGN, DisGeNET, CellMarker, and CMAP. WikiPathways can be replaced by WP, Reactome can be replaced by RP, and CellMarker can be replaced by CM.
organism	Specify species, currently support only Human and Mouse.
GO.ont	GO parameter. One of "BP", "MF", and "CC" subontologies, or "ALL" for all three.
GO.simplify	GO parameter. Whether to remove redundancy of enriched GO terms.
KEGG.use.internal.data	KEGG parameter. Logical, use KEGG.db or latest online KEGG data.
MsigDB.category	MsigDB parameter. MSigDB collection abbreviation, such as All, H, C1, C2, C3, C4, C5, C6, C7.
CMAP.min.Geneset.Size	CMAP parameter. Minimal size of CMAP genes annotated for testing. Recommended use 3.
pvalue.cutoff	pvalue cutoff on enrichment tests to report as significant.
qvalue.cutoff	qvalue cutoff on enrichment tests to report as significant.
padjust.method	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
min.Geneset.Size	Minimal size of genes annotated for testing. Not suitable for CMAP.
max.Geneset.Size	Maximal size of genes annotated for testing.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
genes <- c("CANX", "HSPA1B", "KLRC2", "PSMC6", "RFXAP", "TAP1")
res <- lzq_ORA(genes, enrich.type = "GO")
```

`lzq_ORA.barplot1` *Enrichment barplot for one ORA enrichment object*

Description

Plot enrichment barplot for one ORA enrichment object.

Usage

```
lzq_ORA.barplot1(
  enrich.obj,
  x = "GeneRatio",
  show.term.num = 15,
  color.by = "p.adjust",
  colors = rev(cols_brown_green),
  color.title = color.by,
  bar.width = 0.6,
  add.bar.border = F,
  y.label.position = "right",
  title = NULL,
  legend.position = "right",
  theme.plot = theme_bw(base_rect_size = 1.5),
  use.Chinese = F,
  appid = "20231122001888718",
  key = "5GpDqe8F3pmXfnOkEKGQ"
)
```

Arguments

`enrich.obj`	An object from clusterProfiler.
`x`	variable for x-axis, one of 'GeneRatio', 'pvalue', 'p.adjust', 'Count', EnrichmentFactor.
`show.term.num`	A number or a list of terms. If it is a number, the first n terms will be displayed. If it is a list of terms, the selected terms will be displayed.
`color.by`	variable that used to color enriched terms, one of 'GeneRatio', 'pvalue', 'p.adjust', 'Count', EnrichmentFactor.
`colors`	A color vector for the bars.
`color.title`	Title of color annotation legend.
`bar.width`	Width of bars.
`add.bar.border`	Logical. Whether to add the black border of bars.
`y.label.position`	Y label position. right or left.
`title`	Title of the plot.
`legend.position`	Position of legend. 'none', 'right', 'left' or two numeric variables.
`theme.plot`	ggtheme of plot.
`use.Chinese`	Logical. Whether to use Chinese annotation in the barplot.
`appid`	User app id from baidu translation api. https://fanyi-api.baidu.com/manage/developer.
`key`	User Key from baidu translation api. https://fanyi-api.baidu.com/manage/developer.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
genes <- c("CANX", "HSPA1B", "KLRC2", "PSMC6", "RFXAP", "TAP1")
res <- lzq_ORA(genes, enrich.type = "GO")
lzq_ORA.barplot1(res$simplyGO)
```

---

lzq_ORA.barplot2 *Enrichment barplot for two ORA enrichment objects*

Description

Plot enrichment barplot for two ORA enrichment objects.

Usage

```
lzq_ORA.barplot2(
  enrich.obj1,
  enrich.obj2,
  Selct.P = "FDR",
  cutoff.P = 0.05,
  obj.types = c("Up", "Down"),
  obj.type.colors = c("#ED6355", "#3E94B5"),
  obj1.top.pathway.num = 10,
  obj2.top.pathway.num = 10,
  bar.width = 0.6,
  add.bar.border = T,
  x.limit.fold = 1.05,
  label.size = 3.5,
  legend.position = "bottom",
  use.Chinese = F,
  appid = "20231122001888718",
  key = "5GpDqe8F3pmXfnOkEKGQ"
)
```

Arguments

enrich.obj1	An object from clusterProfiler.
enrich.obj2	An object from clusterProfiler.
Selct.P	Nominal P value (NP) or adjust P value (FDR) were selected to define significant terms.
cutoff.P	A cutoff value for Select_P.
obj.types	Two characters for defining the types of two objects.
obj.type.colors	Two colors for the types of two objects.
obj1.top.pathway.num	The number of top pathways in object 1. Based on the significant test.

obj2.top.pathway.num	The number of top pathways in object 2. Based on the significant test.
bar.width	Width of bar in the plot.
add.bar.border	Logical. Whether to add the black border of bars.
x.limit.fold	Specify the fold of x limitation. Because some terms is too long.
label.size	Fontsize of label.
legend.position	none, left, right, top, bottom; Or Two numeric variables indicated x and y positions, respectively.
use.Chinese	Logical. Whether to use Chinese annotation in the barplot.
appid	User app id from baidu translation api. https://fanyi-api.baidu.com/manage/developer.
key	User Key from baidu translation api. https://fanyi-api.baidu.com/manage/developer.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
library(airway)
library(DESeq2)
library(tidyverse)
library(clusterProfiler)
library(org.Hs.eg.db)
data(airway)
se <- airway
se$dex <- relevel(se$dex, "untrt")
res <- DESeqDataSet(se, design = ~ cell + dex) %>%
  estimateSizeFactors() %>%
  DESeq() %>%
  results() %>%
  as.data.frame() %>%
  na.omit()
ann <- bitr(rownames(res), "ENSEMBL", "SYMBOL", org.Hs.eg.db)
res <- merge(ann, res, by.x = 1, by.y = 0) %>% distinct(SYMBOL, .keep_all = T) # Very crude, just as an example

# Define an up-regulated gene list
up.genes <- res$SYMBOL[res$log2FoldChange > 2 & res$padj < 0.05]
# Define a down-regulated gene list
down.genes <- res$SYMBOL[res$log2FoldChange < -2 & res$padj < 0.05]

# Integrative enrichment analysis of the up-regulated gene list
up.enrich <- lzq_ORA.integrated(genes = up.genes)

# Integrative enrichment analysis of the down-regulated gene list
down.enrich <- lzq_ORA.integrated(genes = down.genes)

lzq_ORA.barplot2(
  enrich.obj1 = up.enrich$simplyGO,
  enrich.obj2 = down.enrich$simplyGO,
  obj.types = c("Up", "Down")
)
```

`lzq_ORA.dotplot1` *Enrichment dotplot for one ORA enrichment object*

Description

Plot enrichment dotplot for one ORA enrichment object.

Usage

```
lzq_ORA.dotplot1(
  enrich.obj,
  x = "GeneRatio",
  show.term.num = 15,
  color.by = "p.adjust",
  colors = rev(cols_brown_green),
  color.title = color.by,
  size.by = "Count",
  size.range = c(3, 8),
  size.title = size.by,
  y.label.position = "right",
  title = NULL,
  legend.position = "bottom",
  theme.plot = theme_bw(base_rect_size = 1.5),
  use.Chinese = F,
  appid = "20231122001888718",
  key = "5GpDqe8F3pmXfnOkEKGQ"
)
```

Arguments

`enrich.obj`	An object from clusterProfiler.
`x`	variable for x-axis, one of 'GeneRatio', 'pvalue', 'p.adjust', 'Count', EnrichmentFactor.
`show.term.num`	A number or a list of terms. If it is a number, the first n terms will be displayed. If it is a list of terms, the selected terms will be displayed.
`color.by`	variable that used to color enriched terms, one of 'GeneRatio', 'pvalue', 'p.adjust', 'Count', EnrichmentFactor.
`colors`	A color vector for the bars.
`color.title`	Title of color annotation legend.
`size.by`	variable that used to size enriched terms, one of 'GeneRatio', 'pvalue', 'p.adjust', 'Count', EnrichmentFactor.
`size.range`	Two numeric variables, the first is minimal value and the first is maximal value.
`size.title`	Title of size annotation legend.
`y.label.position`	Y label position. right or left.
`title`	Title of the plot.
`legend.position`	Postion of legend. 'none', 'right', 'left' or two numeric variables.

theme.plot	ggtheme of plot.
use.Chinese	Logical. Whether to use Chinese annotation in the barplot.
appid	User app id from baidu translation api. https://fanyi-api.baidu.com/manage/developer.
key	User Key from baidu translation api. https://fanyi-api.baidu.com/manage/developer.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
genes <- c("CANX", "HSPA1B", "KLRC2", "PSMC6", "RFXAP", "TAP1")
res <- lzq_ORA(genes, enrich.type = "GO")
lzq_ORA.dotplot1(res$simplyGO)
```

---

lzq_ORA.integrated *Integrate over-representative analysis*

Description

Perform integrated over-representative analysis included GO, KEGG, WikiPathways, Reactome, MsigDB, Disease Ontoloty, Cancer Gene Network, DisGeNET, CellMarker, and CMAP.

Usage

```
lzq_ORA.integrated(
  genes,
  background.genes = NULL,
  gene.type = "SYMBOL",
  organism = "Human",
  GO.ont = "BP",
  KEGG.use.internal.data = F,
  perform.WikiPathways = F,
  perform.Reactome = F,
  perform.MsigDB = F,
  MsigDB.category = "H",
  perform.disease.ontoloty = F,
  perform.Cancer.Gene.Network = F,
  perform.DisGeNET = F,
  perform.CellMarker = F,
  perform.CMAP = T,
  pvalue.cutoff = 0.05,
  qvalue.cutoff = 0.05,
  padjust.method = "BH",
  min.Geneset.Size = 10,
  max.Geneset.Size = 1000,
  CMAP.min.Geneset.Size = 3
)
```

Arguments

genes	A vector of gene id.
background.genes	Background genes. If missing, the all genes listed in the database (eg TERM2GENE table) will be used as background.
gene.type	Keytype of input gene.
organism	Specify species, currently support only Human and Mouse.
GO.ont	One of "BP", "MF", and "CC" subontologies, or "ALL" for all three.
KEGG.use.internal.data	Logical, use KEGG.db or latest online KEGG data.
perform.WikiPathways	Whether to perform WikiPathways enrichment.
perform.Reactome	Whether to perform Reactome enrichment.
perform.MsigDB	Whether to perform MsigDB enrichment.
MsigDB.category	MSigDB collection abbreviation, such as All, H, C1, C2, C3, C4, C5, C6, C7.
perform.disease.ontoloty	Whether to perform DO enrichment.
perform.Cancer.Gene.Network	Whether to perform CGN enrichment.
perform.DisGeNET	Whether to perform DisGeNET enrichment.
perform.CellMarker	Whether to perform CellMarker enrichment. Marker from cellmarker database.
perform.CMAP	Whether to perform CMAP enrichment. Marker from CMAP database (in DSEATM tool).
pvalue.cutoff	pvalue cutoff on enrichment tests to report as significant.
qvalue.cutoff	qvalue cutoff on enrichment tests to report as significant.
padjust.method	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
min.Geneset.Size	Minimal size of genes annotated for testing. Not suitable for CMAP.
max.Geneset.Size	Maximal size of genes annotated for testing.
CMAP.min.Geneset.Size	Minimal size of CMAP genes annotated for testing. Recommended use 3.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
library(airway)
library(DESeq2)
library(tidyverse)
library(clusterProfiler)
library(org.Hs.eg.db)
```

```
data(airway)
se <- airway
se$dex <- relevel(se$dex, "untrt")
res <- DESeqDataSet(se, design = ~ cell + dex) %>%
  estimateSizeFactors() %>%
  DESeq() %>%
  results() %>%
  as.data.frame() %>%
  na.omit()
ann <- bitr(rownames(res), "ENSEMBL", "SYMBOL", org.Hs.eg.db)
res <- merge(ann, res, by.x = 1, by.y = 0) %>% distinct(SYMBOL, .keep_all = T) # Very crude, just as an example

# Define an up-regulated gene list
up.genes <- res$SYMBOL[res$log2FoldChange > 2 & res$padj < 0.05]
# Define a down-regulated gene list
down.genes <- res$SYMBOL[res$log2FoldChange < -2 & res$padj < 0.05]

# Integrative enrichment analysis of the up-regulated gene list
# up.enrich <- lzq_ORA.integrated(genes = up.genes)

# Integrative enrichment analysis of the down-regulated gene list
# down.enrich <- lzq_ORA.integrated(genes = down.genes)
```

`lzq_progeny` *Perform PROGENy analysis*

Description

Perform PROGENy analysis.

Usage

```
lzq_progeny(
  exp,
  scale = T,
  organism = "Human",
  top = 100,
  perm = 1,
  z_scores = F,
  get_nulldist = F,
  assay_name = "RNA",
  return_assay = F
)
```

Arguments

`exp`	A gene expression object with HGNC/MGI symbols in rows and samples in columns. In order to run PROGENy in single-cell RNAseq data, it also accepts Seurat and SingleCellExperiment object, taking the normalized counts for the computation.
`scale`	A logical value indicating whether to scale the scores of each pathway to have a mean of zero and a standard deviation of one. It does not apply if we use permutations.

organism	The model organism - "Human" or "Mouse".
top	The top n genes for generating the model matrix according to significance (p-value).
perm	An interger detailing the number of permutations. No permutations by default (1). When Permutations larger than 1, we compute progeny pathway scores and assesses their significance using a gene sampling-based permutation strategy, for a series of experimental samples/contrasts.
z_scores	Only applies if the number of permutations is greater than 1. A logical value. TRUE: the z-scores will be returned for the pathway activity estimations. FALSE: the function returns a normalized z-score value between -1 and 1.
get_nulldist	Only applies if the number of permutations is greater than 1. A logical value. TRUE: the null distributions generated to assess the signifance of the pathways scores is also returned.
assay_name	Only applies if the input is a Seurat object. It selects the name of the assay on which Progeny will be run. Default to: RNA, i.e. normalized expression values.
return_assay	Only applies if the input is a Seurat object. A logical value indicating whether to return progeny results as a new assay called Progeny in the Seurat object used as input. Default to FALSE.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
gene_expression <- as.matrix(read.csv(system.file("extdata", "human_input.csv", package = "progeny"), row.nam
s <- lzq_progeny(gene_expression)
```

---

lzq_progeny.dea *Perform differential analysis for the PROGENy results.*

Description

Perform differential analysis for the PROGENy results.

Usage

```
lzq_progeny.dea(
  progeny.res,
  groups,
  control.group,
  theme.plot = theme_classic(base_line_size = 0.8)
)
```

Arguments

progeny.res	A PROGENy results with row samples and column pathways.
groups	Group information that matches the counts column sample names.
control.group	Specify the control group.
theme.plot	ggtheme of plot.

Examples

```
gene_expression <- as.matrix(read.csv(system.file("extdata", "human_input.csv", package = "progeny"),
  row.names = 1
))
s <- lzq_progeny(gene_expression)
l <- lzq_progeny.dea(s, groups = rep(c("A", "B"), each = 4), "A")
```

`lzq_progeny.gene.details` *Generate the differential expression and PROGENy weight of genes.*

Description

Generate the differential expression and PROGENy weight of genes.

Usage

```
lzq_progeny.gene.details(
  dea.table,
  pathway,
  organism = "Human",
  top = 100,
  y.lab = bquote(~Log[2] ~ "(Fold change)"),
  colors = c("#3E94B5", "grey70", "#ED6355"),
  point.size = 2,
  label.size = 4,
  theme.plot = theme_classic(base_line_size = 0.8)
)
```

Arguments

`dea.table`	A dataframe with two columns, the first is gene id and the second is logFC/-logP/Stat.
`pathway`	Specific a pathway, such as Androgen, EGFR, Estrogen, Hypoxia, JAK-STAT, MAPK, NFkB, p53, PI3K, TGFb, TNFa, Trail, VEGF, and WNT.
`organism`	The model organism - "Human" or "Mouse".
`top`	The top n genes for generating the model matrix according to significance (p-value).
`y.lab`	Y label for scatter.
`colors`	Colors for different types of points.
`point.size`	Size of point.
`label.size`	Size of gene label.
`theme.plot`	ggtheme of plot.

Examples

```
gene_expression <- as.matrix(read.csv(system.file("extdata", "human_input.csv", package = "progeny"), row.nam
s <- lzq_progeny(gene_expression)
```

lzq_ssGSES *Generate single-sample gene-set enrichment score*

Description

Estimate gene-set enrichment score across all samples.

Usage

```
lzq_ssGSES(exp, gene.list, method = "ssgsea")
```

Arguments

exp	Numeric matrix containing the expression data or gene expression signatures, with samples in columns and genes in rows.
gene.list	Gene sets provided either as a list object or as a GeneSetCollection object.
method	Method to employ in the estimation of gene-set enrichment scores per sample. By default this is set to gsva (Hänzelmann et al, 2013) and other options are ssgsea (Barbie et al, 2009), zscore (Lee et al, 2008) or plage (Tomfohr et al, 2005). The latter two standardize first expression profiles into z-scores over the samples and, in the case of zscore, it combines them together as their sum divided by the square-root of the size of the gene set, while in the case of plage they are used to calculate the singular value decomposition (SVD) over the genes in the gene set and use the coefficients of the first right-singular vector as pathway activity profile.

Value

A gene-set by sample matrix (of matrix or dgCMatrix type, depending on the input) of gene-set enrichment scores.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
gene_expression <- as.matrix(read.csv(system.file("extdata", "human_input.csv", package = "progeny"),
  row.names = 1
))
gl <- list(A = rownames(gene_expression)[1:10], B = rownames(gene_expression)[11:20])
s <- lzq_ssGSES(gene_expression, gl, method = "gsva")
```

lzq_translate *Baidu translation*

Description

Perform Baidu translation.

Usage

```
lzq_translate(
  sentence,
  from = "en",
  to = "zh",
  appid = "20231122001888718",
  key = "5GpDqe8F3pmXfnOkEKGQ"
)
```

Arguments

sentence	A sentence or word need to be translated.
from	Input language type.
to	Output language type.
appid	User app id from baidu translation api. https://fanyi-api.baidu.com/manage/developer.
key	User Key from baidu translation api. https://fanyi-api.baidu.com/manage/developer.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
lzq_translate("BioEnricher is a simple and useful package!")
```

lzq_updateSymbol *Title Identify outdated or Excel-mogrified gene symbols for a gene vector*

Description

Title Identify outdated or Excel-mogrified gene symbols for a gene vector

Usage

```
lzq_updateSymbol(genes, unmapGene_keep = F)
```

Arguments

genes	A gene vector.
unmapGene_keep	whether to keep unmapped genes.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

Examples

```
lzq_updateSymbol("PD-L1")
```

---

lzq_updateSymbolforDL *Title Identify outdated or Excel-mogrified gene symbols for a dataframe*

Description

Title Identify outdated or Excel-mogrified gene symbols for a dataframe

Usage

```
lzq_updateSymbolforDL(data, unmapGene_keep = F)
```

Arguments

data	A expression dataframe with genename rows and sample columns.
unmapGene_keep	whether to keep unmapped genes.

Author(s)

Zaoqu Liu; E-mail: liuzaoqu@163.com

---

searchGEO *Searching GEO metadata*

Description

Searching GEO metadata based on the input term. This function is from genekitr package.

Usage

```
searchGEO(searchterm, minnum = 0, maxnum = 1000)
```

Arguments

searchterm	input searching terms as GEO database keywords, multiple terms are seperated by blanks
minnum	The minimum return records, default is 0
maxnum	The maximum return records, default is 1000

Value

A data.frame.

Examples

```
meta <- searchGEO("ezh2 knockout", maxnum = 5)
```

`searchPubmed` *Get 'PubMed' paper records by searching abstract*

Description

Get 'PubMed' paper records by searching abstract. This function is from europepmc package.

Usage

```
searchPubmed(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 dataframe for PubMed records

Examples

```
term <- c("Tp53", "Brca1", "Tet2")
add_term <- c("stem cell", "mouse")
l <- searchPubmed(term, add_term, num = 30)
```

`searchPubmedTrend` *Get the yearly number of hits for a query and the total yearly number of hits for a given period*

Description

Get the yearly number of hits for a query and the total yearly number of hits for a given period. This function is from europepmc package.

Usage

```
searchPubmedTrend(term, add_term = NULL, period)
```

Arguments

`term`	query terms e.g. gene id, GO/KEGG pathway
`add_term`	other searching terms Default is NULL
`period`	a vector of years (numeric) over which to perform the search.

Value

a data.frame (dplyr tbl_df) with year, total number of hits (all_hits) and number of hits for the query (query_hits).

Examples

```
term <- c("Tp53", "Brca1", "Tet2")
add_term <- c("stem cell", "mouse")
l <- searchPubmedTrend(term, add_term, period = 2020:2023)
```

Index