Package 'clusterProfiler'

November 11, 2026

Type Package

Title A Universal Enrichment Tool for Interpreting Omics Data

Version 4.20.3

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Description A universal tool for interpreting functional characteristics of omics data.
It supports Over-Representation Analysis (ORA) and Gene Set Enrichment Analysis (GSEA) for both coding and non-coding genomics data of thousands of species.
It provides a unified and tidy interface to access, manipulate, and visualize enrichment results.
A key capability is the simultaneous analysis and comparison of datasets from multiple treatments or time points.
Furthermore, it integrates Large Language Model (LLM) capabilities to provide automated and insightful interpretation of enrichment results.

Depends R (>= 4.2.0)

Imports aisdk, AnnotationDbi, dplyr, enrichit (>= 0.1.1), enrichplot (>= 1.9.3), ggplot2, GO.db, GOSemSim (>= 2.27.2), gson (>= 0.0.7), httr, igraph, jsonlite, magrittr, methods, plyr, qvalue, rlang, stats, tidyr, utils, yulab.utils (>= 0.2.3)

Suggests AnnotationHub, BiocManager, DOSE, ggtangle, readr, org.Hs.eg.db, quarto, testthat

VignetteBuilder quarto

ByteCompile true

License Artistic-2.0

URL https://yulab-smu.top/contribution-knowledge-mining/

BugReports https://github.com/YuLab-SMU/clusterProfiler/issues

biocViews Annotation, Clustering, GeneSetEnrichment, GO, KEGG, MultipleComparison, Pathways, Reactome, Visualization

Encoding UTF-8

Config/roxygen2/version 8.1.1

git_url https://git.bioconductor.org/packages/clusterProfiler

git_branch RELEASE_3_23

git_last_commit 72bb1f4

git_last_commit_date 2026-10-18

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Guangchuang Yu [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-6485-8781>),
Li-Gen Wang [ctb],
Xiao Luo [ctb],
Meijun Chen [ctb],
Giovanni Dall'Olio [ctb],
Wanqian Wei [ctb],
Chun-Hui Gao [ctb] (ORCID: <https://orcid.org/0000-0002-1445-7939>)

Contents

clusterProfiler-package . . . 3
append_kegg_category . . . 4
bitr . . . 5
bitr_kegg . . . 5
browseKEGG . . . 6
compareCluster . . . 6
DataSet . . . 8
download_KEGG . . . 8
dropGO . . . 9
enrichDAVID . . . 9
enricher . . . 10
enrichGO . . . 11
enrichKEGG . . . 13
enrichMKEGG . . . 14
enrichPC . . . 15
enrichWP . . . 16
getPPI . . . 16
getTaxID . . . 18
getTaxInfo . . . 18
get_wp_organisms . . . 19
Gff2GeneTable . . . 19
go2ont . . . 20
go2term . . . 20
gofilter . . . 21
groupGO . . . 21
groupGOResult-class . . . 22
GSEA . . . 23
gseGO . . . 24
gseKEGG . . . 26
gseMKEGG . . . 27
gsePC . . . 28

gseWP 29
gson_GO 30
gson_GO_local 30
gson_KEGG 31
gson_KEGG_mapper 31
gson_WP 32
idType 33
infer_model_id 33
interpret 34
interpret_agent 35
interpret_hierarchical 37
ko2name 38
merge_result 38
plot.interpretation 39
plotGOgraph 39
read.gmt.pc 40
reexports 40
search_kegg_organism 41
simplify 41
uniprot_get 43

Index **44**

`clusterProfiler-package` *clusterProfiler: A Universal Enrichment Tool for Interpreting Omics Data*

Description

A universal tool for interpreting functional characteristics of omics data. It supports Over-Representation Analysis (ORA) and Gene Set Enrichment Analysis (GSEA) for both coding and non-coding genomics data of thousands of species. It provides a unified and tidy interface to access, manipulate, and visualize enrichment results. A key capability is the simultaneous analysis and comparison of datasets from multiple treatments or time points. Furthermore, it integrates Large Language Model (LLM) capabilities to provide automated and insightful interpretation of enrichment results.

Author(s)

Maintainer: Guangchuang Yu <guangchuangyu@gmail.com> (ORCID) [copyright holder]

Authors:

- Guangchuang Yu <guangchuangyu@gmail.com> (ORCID) [copyright holder]

Other contributors:

- Li-Gen Wang <reeganwang020@gmail.com> [contributor]
- Xiao Luo <l77880853349@163.com> [contributor]

- Meijun Chen <mjchen1996@outlook.com> [contributor]
- Giovanni Dall'Olio <giovanni.dallolio@upf.edu> [contributor]
- Wanqian Wei <altair_wei@outlook.com> [contributor]
- Chun-Hui Gao <gaospecial@gmail.com> (ORCID) [contributor]

See Also

Useful links:

- https://yulab-smu.top/contribution-knowledge-mining/
- Report bugs at https://github.com/YuLab-SMU/clusterProfiler/issues

---

`append_kegg_category` *append_kegg_category*

Description

add KEGG pathway category information

Usage

```
append_kegg_category(x)
```

Arguments

x	KEGG enrichment result

Details

This function appends the KEGG pathway category information to KEGG enrichment result (either output of 'enrichKEGG' or 'gseKEGG'

Value

update KEGG enrichment result with category information

Author(s)

Guangchuang Yu

bitr *bitr*

Description

Biological Id TRanslator

Usage

```
bitr(geneID, fromType, toType, OrgDb, drop = TRUE)
```

Arguments

geneID	input gene id
fromType	input id type
toType	output id type
OrgDb	annotation db
drop	drop NA or not

Value

data.frame

Author(s)

Guangchuang Yu

bitr_kegg *bitr_kegg*

Description

convert biological ID using KEGG API

Usage

```
bitr_kegg(geneID, fromType, toType, organism, drop = TRUE)
```

Arguments

geneID	input gene id
fromType	input id type
toType	output id type
organism	supported organism, can be search using search_kegg_organism function
drop	drop NA or not

Value

data.frame

Author(s)

Guangchuang Yu

`browseKEGG` *browseKEGG*

Description

open KEGG pathway with web browser

Usage

```
browseKEGG(x, pathID)
```

Arguments

`x`	an instance of enrichResult or gseaResult
`pathID`	pathway ID

Value

url

Author(s)

Guangchuang Yu

`compareCluster` *Compare gene clusters functional profile*

Description

Given a list of gene set, this function will compute profiles of each gene cluster.

Usage

```
compareCluster(
  geneClusters,
  fun = "enrichGO",
  data = "",
  source_from = NULL,
  ...
)
```

Arguments

geneClusters	a list of entrez gene id. Alternatively, a formula of type Entrez~group or a formula of type Entrez | logFC ~ group for "gseGO", "gseKEGG" and "GSEA".
fun	One of "groupGO", "enrichGO", "enrichKEGG", "enrichDO" or "enrichPathway" . Users can also supply their own function.
data	if geneClusters is a formula, the data from which the clusters must be extracted.
source_from	If using a custom function in "fun", provide the source package as a string here. Otherwise, the function will be obtained from the global environment.
...	Other arguments.

Value

A clusterProfResult instance.

Author(s)

Guangchuang Yu https://yulab-smu.top

See Also

[compareClusterResult-class], [groupGO], [enrichGO], [enrichKEGG], [enrichDO][DOSE::enrichDO], [enrichPathway][ReactomePA::enrichPathway]

Examples

```
## Not run:
data(gcSample)
xx <- compareCluster(gcSample, fun="enrichKEGG",
                     organism="hsa", pvalueCutoff=0.05)
as.data.frame(xx)
# plot(xx, type="dot", caption="KEGG Enrichment Comparison")
dotplot(xx)

## formula interface
mydf <- data.frame(Entrez=c('1', '100', '1000', '100101467',
                            '100127206', '100128071'),
                   logFC = c(1.1, -0.5, 5, 2.5, -3, 3),
                   group = c('A', 'A', 'A', 'B', 'B', 'B'),
                   othergroup = c('good', 'good', 'bad', 'bad', 'good', 'bad'))
xx.formula <- compareCluster(Entrez~group, data=mydf,
                             fun='groupGO', OrgDb='org.Hs.eg.db')
as.data.frame(xx.formula)

## formula interface with more than one grouping variable
xx.formula.twogroups <- compareCluster(Entrez~group+othergroup, data=mydf,
                                       fun='groupGO', OrgDb='org.Hs.eg.db')
as.data.frame(xx.formula.twogroups)


## End(Not run)
```

DataSet *Datasets gcSample contains a sample of gene clusters.*

Description

Datasets gcSample contains a sample of gene clusters.

Datasets kegg_species contains kegg species information

Datasets kegg_category contains kegg pathway category information

Datasets DE_GSE8057 contains differential epxressed genes obtained from GSE8057 dataset

download_KEGG *download_KEGG*

Description

download the latest version of KEGG pathway/module

Usage

```
download_KEGG(species, keggType = "KEGG", keyType = "kegg")
```

Arguments

species	species
keggType	one of 'KEGG' or 'MKEGG'
keyType	supported keyType, see bitr_kegg

Value

list

Author(s)

Guangchuang Yu

dropGO *dropGO*

Description

drop GO term of specific level or specific terms (mostly too general).

Usage

```
dropGO(x, level = NULL, term = NULL)
```

Arguments

x	an instance of 'enrichResult' or 'compareClusterResult'
level	GO level
term	GO term

Value

modified version of x

Author(s)

Guangchuang Yu

---

enrichDAVID *enrichDAVID*

Description

enrichment analysis by DAVID

Usage

```
enrichDAVID(
  gene,
  idType = "ENTREZ_GENE_ID",
  universe,
  minGSSize = 10,
  maxGSSize = 500,
  annotation = "GOTERM_BP_FAT",
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  qvalueCutoff = 0.2,
  species = NA,
  david.user
)
```

Arguments

gene	input gene
idType	id type
universe	background genes. If missing, the all genes listed in the database (eg TERM2GENE table) will be used as background.
minGSSize	minimal size of genes annotated for testing
maxGSSize	maximal size of genes annotated for testing
annotation	david annotation
pvalueCutoff	adjusted pvalue cutoff on enrichment tests to report
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"
qvalueCutoff	qvalue cutoff on enrichment tests to report as significant. Tests must pass i) pvalueCutoff on unadjusted pvalues, ii) pvalueCutoff on adjusted pvalues and iii) qvalueCutoff on qvalues to be reported.
species	species
david.user	david user

Value

A enrichResult instance

Author(s)

Guangchuang Yu

---

enricher *enricher*

Description

A universal enrichment analyzer

Usage

```
enricher(
  gene,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  universe = NULL,
  minGSSize = 10,
  maxGSSize = 500,
  qvalueCutoff = 0.2,
  gson = NULL,
  TERM2GENE,
  TERM2NAME = NA
)
```

Arguments

gene	a vector of gene id
pvalueCutoff	adjusted pvalue cutoff on enrichment tests to report
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"
universe	background genes. If missing, the all genes listed in the database (eg TERM2GENE table) will be used as background.
minGSSize	minimal size of genes annotated for testing
maxGSSize	maximal size of genes annotated for testing
qvalueCutoff	qvalue cutoff on enrichment tests to report as significant. Tests must pass i) pvalueCutoff on unadjusted pvalues, ii) pvalueCutoff on adjusted pvalues and iii) qvalueCutoff on qvalues to be reported.
gson	a GSON object, if not NULL, use it as annotation data.
TERM2GENE	user input annotation of TERM TO GENE mapping, a data.frame of 2 column with term and gene. Only used when gson is NULL.
TERM2NAME	user input of TERM TO NAME mapping, a data.frame of 2 column with term and name. Only used when gson is NULL.

Value

A enrichResult instance

Author(s)

Guangchuang Yu https://yulab-smu.top

---

enrichGO *GO Enrichment Analysis of a gene set. Given a vector of genes, this function will return the enrichment GO categories after FDR control.*

Description

GO Enrichment Analysis of a gene set. Given a vector of genes, this function will return the enrichment GO categories after FDR control.

Usage

```
enrichGO(
  gene,
  OrgDb,
  keyType = "ENTREZID",
  ont = "MF",
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  universe,
```

```
  qvalueCutoff = 0.2,
  minGSSize = 10,
  maxGSSize = 500,
  readable = FALSE,
  pool = FALSE
)
```

Arguments

gene	a vector of entrez gene id.
OrgDb	OrgDb
keyType	keytype of input gene
ont	One of "BP", "MF", and "CC" subontologies, or "ALL" for all three.
pvalueCutoff	adjusted pvalue cutoff on enrichment tests to report
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"
universe	background genes. If missing, the all genes listed in the database (eg TERM2GENE table) will be used as background.
qvalueCutoff	qvalue cutoff on enrichment tests to report as significant. Tests must pass i) pvalueCutoff on unadjusted pvalues, ii) pvalueCutoff on adjusted pvalues and iii) qvalueCutoff on qvalues to be reported.
minGSSize	minimal size of genes annotated by Ontology term for testing.
maxGSSize	maximal size of genes annotated for testing
readable	whether mapping gene ID to gene Name
pool	If ont='ALL', whether pool 3 GO sub-ontologies

Value

An enrichResult instance.

Author(s)

Guangchuang Yu https://yulab-smu.top

See Also

[enrichResult-class], [compareCluster]

Examples

```
## Not run:
  data(geneList, package = "DOSE")
 de <- names(geneList)[1:100]
 yy <- enrichGO(de, 'org.Hs.eg.db', ont="BP", pvalueCutoff=0.01)
 head(yy)

## End(Not run)
```

enrichKEGG *KEGG Enrichment Analysis of a gene set. Given a vector of genes, this function will return the enrichment KEGG categories with FDR control.*

Description

KEGG Enrichment Analysis of a gene set. Given a vector of genes, this function will return the enrichment KEGG categories with FDR control.

Usage

```
enrichKEGG(
  gene,
  organism = "hsa",
  keyType = "kegg",
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  universe,
  minGSSize = 10,
  maxGSSize = 500,
  qvalueCutoff = 0.2,
  use_internal_data = FALSE
)
```

Arguments

gene	a vector of entrez gene id.
organism	supported organism listed in 'https://www.genome.jp/kegg/catalog/org_list.html'
keyType	one of "kegg", 'ncbi-geneid', 'ncbi-proteinid' and 'uniprot'
pvalueCutoff	adjusted pvalue cutoff on enrichment tests to report
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"
universe	background genes. If missing, the all genes listed in the database (eg TERM2GENE table) will be used as background.
minGSSize	minimal size of genes annotated by Ontology term for testing.
maxGSSize	maximal size of genes annotated for testing
qvalueCutoff	qvalue cutoff on enrichment tests to report as significant. Tests must pass i) pvalueCutoff on unadjusted pvalues, ii) pvalueCutoff on adjusted pvalues and iii) qvalueCutoff on qvalues to be reported.
use_internal_data	logical, use KEGG.db or latest online KEGG data

Value

A enrichResult instance.

Author(s)

Guangchuang Yu https://yulab-smu.top

See Also

[enrichResult-class], [compareCluster]

Examples

```
## Not run:
  data(geneList, package='DOSE')
  de <- names(geneList)[1:100]
  yy <- enrichKEGG(de, pvalueCutoff=0.01)
  head(yy)

## End(Not run)
```

`enrichMKEGG` *KEGG Module Enrichment Analysis of a gene set. Given a vector of genes, this function will return the enrichment KEGG Module categories with FDR control.*

Description

KEGG Module Enrichment Analysis of a gene set. Given a vector of genes, this function will return the enrichment KEGG Module categories with FDR control.

Usage

```
enrichMKEGG(
  gene,
  organism = "hsa",
  keyType = "kegg",
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  universe,
  minGSSize = 10,
  maxGSSize = 500,
  qvalueCutoff = 0.2
)
```

Arguments

`gene`	a vector of entrez gene id.
`organism`	supported organism listed in 'https://www.genome.jp/kegg/catalog/org_list.html'
`keyType`	one of "kegg", 'ncbi-geneid', 'ncbi-proteinid' and 'uniprot'
`pvalueCutoff`	adjusted pvalue cutoff on enrichment tests to report

pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"
universe	background genes. If missing, the all genes listed in the database (eg TERM2GENE table) will be used as background.
minGSSize	minimal size of genes annotated by Ontology term for testing.
maxGSSize	maximal size of genes annotated for testing
qvalueCutoff	qvalue cutoff on enrichment tests to report as significant. Tests must pass i) pvalueCutoff on unadjusted pvalues, ii) pvalueCutoff on adjusted pvalues and iii) qvalueCutoff on qvalues to be reported.

Value

A enrichResult instance.

---

enrichPC *enrichPC*

Description

ORA analysis for Pathway Commons

Usage

```
enrichPC(gene, ...)
```

Arguments

gene	a vector of genes (either hgnc symbols or uniprot IDs)
...	additional parameters, see also the parameters supported by the enricher() function

Details

This function performs over-representation analysis using Pathway Commons

Value

A enrichResult instance

enrichWP *enrichWP*

Description

ORA analysis for WikiPathways

Usage

```
enrichWP(gene, organism, ...)
```

Arguments

gene	a vector of entrez gene id
organism	supported organisms, which can be accessed via the get_wp_organisms() function
...	additional parameters, see also the parameters supported by the enricher() function

Details

This function performs over-representation analysis using WikiPathways

Value

A enrichResult instance

Author(s)

Guangchuang Yu

getPPI *getPPI*

Description

getPPI

Usage

```
getPPI(
  x,
  ID = 1,
  taxID = "auto",
  required_score = NULL,
  network_type = "functional",
  add_nodes = 0,
  show_query_node_labels = 0,
  output = "igraph"
)
```

Arguments

x	an ‘enrichResult“ object or a vector of proteins, e.g. ‘c("PTCH1", "TP53", "BRCA1", "BRCA2")‘
ID	ID or index to extract genes in the enriched term(s) if ‘x‘ is an ‘enrichResult‘ object
taxID	NCBI taxon identifiers (e.g. Human is 9606, see: [STRING organisms](https://string-db.org/cgi/input.pl?input_page_active_form=organisms).
required_score	threshold of significance to include a interaction, a number between 0 and 1000 (default depends on the network)
network_type	network type: functional (default), physical
add_nodes	adds a number of proteins with to the network based on their confidence score (default:1)
show_query_node_labels	when available use submitted names in the preferredName column when (0 or 1) (default:0)
output	one of ‘data.frame‘ or ‘igraph‘

Details

[Getting the STRING network interactions](https://string-db.org/cgi/help.pl?sessionId=btsvnCeNrBk7).

Value

a ‘data.frame‘ or an ‘igraph‘ object

Author(s)

Yonghe Xia and modified by Guangchuang Yu

getTaxID *getTaxID*

Description

Convert species scientific name to taxonomic ID

Usage

```
getTaxID(species)
```

Arguments

species	scientific name of a species

Value

taxonomic ID

Author(s)

Guangchuang Yu

getTaxInfo *getTaxInfo*

Description

Query taxonomy information from ‘stringdb‘ or ‘ensembl‘ web services

Usage

```
getTaxInfo(species, source = "stringdb")
```

Arguments

species	scientific name of a species
source	one of ‘stringdb‘ or ‘ensembl‘

Value

a ‘data.frame‘ of query information

Author(s)

Guangchuang Yu

get_wp_organisms *get_wp_organism*

Description

list supported organism of WikiPathways

Usage

```
get_wp_organisms()
```

Details

This function extracts information from 'https://data.wikipathways.org/current/gmt/' and lists all supported organisms

Value

supported organism list

Author(s)

Guangchuang Yu

Gff2GeneTable *Gff2GeneTable*

Description

read GFF file and build gene information table

Usage

```
Gff2GeneTable(gffFile, compress = TRUE)
```

Arguments

gffFile	GFF file
compress	compress file or not

Details

given a GFF file, this function extracts information from it and save it in working directory

Value

file save.

Author(s)

Yu Guangchuang

`go2ont` *go2ont*

Description

convert goid to ontology (BP, CC, MF)

Usage

```
go2ont(goid)
```

Arguments

`goid`	a vector of GO IDs

Value

data.frame

Author(s)

Guangchuang Yu

`go2term` *go2term*

Description

convert goid to descriptive term

Usage

```
go2term(goid)
```

Arguments

`goid`	a vector of GO IDs

Value

data.frame

Author(s)

Guangchuang Yu

gofilter *gofilter*

Description

filter GO enriched result at specific level

Usage

```
gofilter(x, level = 4)
```

Arguments

x	output from enrichGO or compareCluster
level	GO level

Value

updated object

Author(s)

Guangchuang Yu

groupGO *Functional Profile of a gene set at specific GO level. Given a vector of genes, this function will return the GO profile at a specific level.*

Description

Functional Profile of a gene set at specific GO level. Given a vector of genes, this function will return the GO profile at a specific level.

Usage

```
groupGO(
  gene,
  OrgDb,
  keyType = "ENTREZID",
  ont = "CC",
  level = 2,
  readable = FALSE
)
```

Arguments

gene	a vector of entrez gene id.
OrgDb	OrgDb
keyType	key type of input gene
ont	One of "MF", "BP", and "CC" subontologies.
level	Specific GO Level.
readable	if readable is TRUE, the gene IDs will mapping to gene symbols.

Value

A groupGOResult instance.

Author(s)

Guangchuang Yu https://yulab-smu.top

See Also

[groupGOResult-class], [compareCluster]

Examples

```
data(gcSample)
yy <- groupGO(gcSample[[1]], 'org.Hs.eg.db', ont="BP", level=2)
head(summary(yy))
#plot(yy)
```

---

groupGOResult-class *Class "groupGOResult" This class represents the result of functional Profiles of a set of gene at specific GO level.*

Description

Class "groupGOResult" This class represents the result of functional Profiles of a set of gene at specific GO level.

Slots

result GO classification result

ontology Ontology

level GO level

organism one of "human", "mouse" and "yeast"

gene Gene IDs

readable logical flag of gene ID in symbol or not.

Author(s)

Guangchuang Yu https://yulab-smu.top

See Also

[compareClusterResult], [compareCluster], [groupGO]

GSEA *GSEA*

Description

a universal gene set enrichment analysis tools

Usage

```
GSEA(
  geneList,
  exponent = 1,
  minGSSize = 10,
  maxGSSize = 500,
  eps = 1e-10,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  gson = NULL,
  TERM2GENE,
  TERM2NAME = NA,
  verbose = TRUE,
  nPerm = 1000,
  method = "multilevel",
  adaptive = FALSE,
  minPerm = 101,
  maxPerm = 1e+05,
  pvalThreshold = 0.1,
  ...
)
```

Arguments

geneList	order ranked geneList
exponent	weight of each step
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of genes annotated for testing
eps	boundary for calculating the p value in multilevel mode
pvalueCutoff	adjusted pvalue cutoff

pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"
gson	a GSON object, if not NULL, use it as annotation data.
TERM2GENE	user input annotation of TERM TO GENE mapping, a data.frame of 2 column with term and gene. Only used when gson is NULL.
TERM2NAME	user input of TERM TO NAME mapping, a data.frame of 2 column with term and name. Only used when gson is NULL.
verbose	logical
nPerm	The number of permutations.
method	method of calculating the pvalue, one of "multilevel", "monte carlo" and "fgsea"
adaptive	logical, whether to use adaptive method for calculating pvalue
minPerm	minimal number of permutations for adaptive method
maxPerm	maximal number of permutations for adaptive method
pvalThreshold	pvalue threshold for adaptive method
...	other parameters passed to `enrichit::gsea_gson()`

Value

gseaResult object

Author(s)

Guangchuang Yu https://yulab-smu.top

`gseGO` *gseGO*

Description

Gene Set Enrichment Analysis of Gene Ontology

Usage

```
gseGO(
  geneList,
  ont = "BP",
  OrgDb,
  keyType = "ENTREZID",
  exponent = 1,
  minGSSize = 10,
  maxGSSize = 500,
  eps = 1e-10,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  verbose = TRUE,
```

```
  nPerm = 1000,
  method = "multilevel",
  adaptive = FALSE,
  minPerm = 101,
  maxPerm = 1e+05,
  pvalThreshold = 0.1,
  ...
)
```

Arguments

geneList	order ranked geneList
ont	one of "BP", "MF", and "CC" subontologies, or "ALL" for all three.
OrgDb	OrgDb
keyType	keytype of gene
exponent	weight of each step
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of genes annotated for testing
eps	boundary for calculating the p value in multilevel mode
pvalueCutoff	pvalue Cutoff
pAdjustMethod	pvalue adjustment method
verbose	print message or not
nPerm	The number of permutations.
method	method of calculating the pvalue, one of "multilevel", "monte carlo" and "fgsea"
adaptive	logical, whether to use adaptive method for calculating pvalue
minPerm	minimal number of permutations for adaptive method
maxPerm	maximal number of permutations for adaptive method
pvalThreshold	pvalue threshold for adaptive method
...	other parameters passed to `enrichit::gsea_gson()`

Value

gseaResult object

Author(s)

Yu Guangchuang

gseKEGG *gseKEGG*

Description

Gene Set Enrichment Analysis of KEGG

Usage

```
gseKEGG(
  geneList,
  organism = "hsa",
  keyType = "kegg",
  exponent = 1,
  minGSSize = 10,
  maxGSSize = 500,
  eps = 1e-10,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  verbose = TRUE,
  use_internal_data = FALSE,
  nPerm = 1000,
  method = "multilevel",
  adaptive = FALSE,
  minPerm = 101,
  maxPerm = 1e+05,
  pvalThreshold = 0.1,
  ...
)
```

Arguments

geneList	order ranked geneList
organism	supported organism listed in 'https://www.genome.jp/kegg/catalog/org_list.html'
keyType	one of "kegg", 'ncbi-geneid', 'ncib-proteinid' and 'uniprot'
exponent	weight of each step
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of genes annotated for testing
eps	boundary for calculating the p value in multilevel mode
pvalueCutoff	pvalue Cutoff
pAdjustMethod	pvalue adjustment method
verbose	print message or not
use_internal_data	logical, use KEGG.db or latest online KEGG data

nPerm	The number of permutations.
method	method of calculating the pvalue, one of "multilevel", "monte carlo" and "fgsea"
adaptive	logical, whether to use adaptive method for calculating pvalue
minPerm	minimal number of permutations for adaptive method
maxPerm	maximal number of permutations for adaptive method
pvalThreshold	pvalue threshold for adaptive method
...	other parameters passed to enrichit::gsea_gson()

Value

gseaResult object

Author(s)

Yu Guangchuang

---

gseMKEGG *gseMKEGG*

Description

Gene Set Enrichment Analysis of KEGG Module

Usage

```
gseMKEGG(
  geneList,
  organism = "hsa",
  keyType = "kegg",
  exponent = 1,
  minGSSize = 10,
  maxGSSize = 500,
  eps = 1e-10,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  verbose = TRUE,
  nPerm = 1000,
  method = "multilevel",
  adaptive = FALSE,
  minPerm = 101,
  maxPerm = 1e+05,
  pvalThreshold = 0.1,
  ...
)
```

Arguments

geneList	order ranked geneList
organism	supported organism listed in 'https://www.genome.jp/kegg/catalog/org_list.html'
keyType	one of "kegg", 'ncbi-geneid', 'ncib-proteinid' and 'uniprot'
exponent	weight of each step
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of genes annotated for testing
eps	boundary for calculating the p value in multilevel mode
pvalueCutoff	pvalue Cutoff
pAdjustMethod	pvalue adjustment method
verbose	print message or not
nPerm	The number of permutations.
method	method of calculating the pvalue, one of "multilevel", "monte carlo" and "fgsea"
adaptive	logical, whether to use adaptive method for calculating pvalue
minPerm	minimal number of permutations for adaptive method
maxPerm	maximal number of permutations for adaptive method
pvalThreshold	pvalue threshold for adaptive method
...	other parameters passed to `enrichit::gsea_gson()`

Value

gseaResult object

Author(s)

Yu Guangchuang

---

gsePC *gsePC*

Description

GSEA analysis for Pathway Commons

Usage

```
gsePC(geneList, ...)
```

Arguments

geneList	a ranked gene list
...	additional parameters, see also the parameters supported by the GSEA() function

Details

This function performs GSEA using Pathway Commons

Value

A `gseaResult` instance

`gseWP` *gseWP*

Description

GSEA analysis for WikiPathways

Usage

```
gseWP(geneList, organism, ...)
```

Arguments

`geneList`	ranked gene list
`organism`	supported organisms, which can be accessed via the get_wp_organisms() function
`...`	additional parameters, see also the parameters supported by the GSEA() function

Details

This function performs GSEA using WikiPathways

Value

A `gseaResult` instance

Author(s)

Guangchuang Yu

gson_GO *gson_KEGG*

Description

download the latest version of KEGG pathway and stored in a 'GSON' object

Usage

```
gson_GO(OrgDb, keytype = "ENTREZID", ont = "BP")
```

Arguments

OrgDb	OrgDb
keytype	keytype of genes.
ont	one of "BP", "MF", "CC", and "ALL"

Value

a 'GSON' object

gson_GO_local *Build a gson object that annotate Gene Ontology*

Description

Build a gson object that annotate Gene Ontology

Usage

```
gson_GO_local(data, ont = c("ALL", "BP", "CC", "MF"), species = NULL, ...)
```

Arguments

data	a two-column data.frame of original GO annotation. The columns are "gene_id" and "go_id".
ont	type of GO annotation, which is "ALL", "BP", "MF", or "CC". default: "ALL".
species	name of species. Default: NULL.
...	pass to ‘gson::gson()‘ constructor.

Value

a ‘gson‘ instance

Examples

```
data = data.frame(gene_id = "gene1",
            go_id = c("GO:0035492", "GO:0009764", "GO:0031040", "GO:0033714", "GO:0036349"))
gson_GO_local(data, species = "E. coli")
```

`gson_KEGG` *gson_KEGG*

Description

download the latest version of KEGG pathway and stored in a 'GSON' object

Usage

```
gson_KEGG(species, KEGG_Type = "KEGG", keyType = "kegg")
```

Arguments

`species`	species
`KEGG_Type`	one of "KEGG" and "MKEGG"
`keyType`	one of "kegg", 'ncbi-geneid', 'ncib-proteinid' and 'uniprot'.

Value

a 'GSON' object

Author(s)

Guangchuang Yu

`gson_KEGG_mapper` *Build KEGG annotation for novel species using KEGG Mapper*

Description

KEGG Mapper service can annotate protein sequences for novel species with KO database, and KO annotation need to be converted into Pathway or Module annotation, which can then be used in 'clusterProfiler'

Usage

```
gson_KEGG_mapper(
  file,
  format = c("BLAST", "Ghost", "Kofam"),
  type = c("pathway", "module"),
  species = NULL,
  ...
)
```

Arguments

file	the name of the file which comes from the KEGG Mapper service, see Details for file format
format	string indicate format of KEGG Mapper result
type	string indicate annotation database
species	your species, NULL if ignored
...	pass to gson::gson()

Details

File is a two-column dataset with K numbers in the second column, optionally preceded by the user's identifiers in the first column. This is consistent with the output files of automatic annotation servers, BlastKOALA, GhostKOALA, and KofamKOALA. KOALA (KEGG Orthology And Links Annotation) is KEGG's internal annotation tool for K number assignment of KEGG GENES using SSEARCH computation. BlastKOALA and GhostKOALA assign K numbers to the user's sequence data by BLAST and GHOSTX searches, respectively, against a nonredundant set of KEGG GENES. KofamKOALA is a new member of the KOALA family available at GenomeNet using the HMM profile search, rather than the sequence similarity search, for K number assignment. see https://www.kegg.jp/blastkoala/, https://www.kegg.jp/ghostkoala/ and https://www.genome.jp/tools/kofamkoala/ for more information.

Value

a gson instance

Examples

```
## Not run:
 file = system.file('extdata', "kegg_mapper_blast.txt", package='clusterProfiler')
 gson_KEGG_mapper(file, format = "BLAST", type = "pathway")

## End(Not run)
```

---

gson_WP *gson_WP*

Description

Download the latest version of WikiPathways data and stored in a 'GSON' object

Usage

```
gson_WP(organism)
```

Arguments

organism	supported organism, which can be accessed via the get_wp_organisms() function.

idType *idType*

Description

list ID types supported by annoDb

Usage

```
idType(OrgDb = "org.Hs.eg.db")
```

Arguments

OrgDb	annotation db

Value

character vector

Author(s)

Guangchuang Yu

infer_model_id *Infer Model ID*

Description

Maps bare model names to the aisdk ‘provider:model‘ format for backward compatibility. Emits a warning when guessing and suggests the explicit form. If the model already contains a colon, it is returned as-is.

Usage

```
infer_model_id(model)
```

Arguments

model	A model string, either bare (e.g., "deepseek-chat") or fully qualified (e.g., "deepseek:deepseek-chat").

Value

A string in ‘provider:model‘ format.

`interpret` *Interpret Enrichment Results Using LLMs*

Description

Functions for interpreting functional enrichment analysis results using Large Language Models. Supports single-call interpretation, multi-agent deep analysis, and hierarchical cluster strategies.

Built on top of aisdk's 'generate_object()' for reliable structured output, and the Agent/Session system for multi-agent pipelines.

Sends enrichment results along with optional experimental context to an LLM to generate a structured biological interpretation, hypothesis, and narrative suitable for a publication.

Usage

```
interpret(
  x,
  context = NULL,
  n_pathways = 20,
  model = NULL,
  task = "interpretation",
  prior = NULL,
  add_ppi = FALSE,
  gene_fold_change = NULL,
  max_tokens = 8192,
  temperature = 0.3,
  verbose = FALSE
)
```

Arguments

`x`	An enrichment result object ('enrichResult', 'gseaResult', 'compareClusterResult', or a 'data.frame' with pathway columns).
`context`	A string describing the experimental background (e.g., "scRNA-seq of mouse myocardial infarction at day 3").
`n_pathways`	Number of top significant pathways to include. Default 20.
`model`	Optional LLM model. When 'NULL' (default), uses the aisdk package-wide default model configured via 'aisdk::set_model()'. You can also supply a model ID in 'provider:model' format (e.g., '"deepseek:deepseek-chat"', '"gemini:gemini-2.5-flash"') or a 'LanguageModelV1' object. Bare model names are supported with a warning (e.g., '"deepseek-chat"').
`task`	Task type: "interpretation" (default), "cell_type"/"annotation", or "phenotype"/"phenotyping".
`prior`	Optional prior knowledge or preliminary annotation to guide the task.
`add_ppi`	Logical, whether to query STRING PPI network data. Default FALSE.
`gene_fold_change`	Named numeric vector of log fold changes for expression context.

max_tokens	Maximum tokens for the LLM response. Default 8192. Some models (especially reasoning models) may need much higher values (e.g., 16384 or more) to produce complete structured output.
temperature	Sampling temperature. Default 0.3.
verbose	Logical, whether to print debug messages showing raw API responses, token usage, and JSON parsing details. Default FALSE. Equivalent to setting ‘options(aisdk.debug = TRUE)‘ for the call.

Details

Uses ‘generate_object()‘ internally for reliable structured output with automatic JSON repair, eliminating manual parsing failures.

Value

An ‘interpretation‘ object (list) with task-specific fields. For "interpretation": overview, key_mechanisms, hypothesis, narrative, etc. For "annotation": cell_type, confidence, reasoning, markers, etc. For "phenotype": phenotype, confidence, reasoning, key_processes, etc.

Examples

```
## Not run:
# Basic usage with a data frame
df <- data.frame(
  ID = c("GO:0006915", "GO:0008284"),
  Description = c("apoptotic process", "positive regulation of proliferation"),
  GeneRatio = c("10/100", "20/100"),
  p.adjust = c(0.01, 0.02),
  geneID = c("TP53/BAX", "MYC/CCND1/CDK4")
)
res <- interpret(df,
  model = "deepseek:deepseek-chat",
  context = "Cancer proliferation study"
)
# Reuse aisdk's global default model
# aisdk::set_model("openai:gpt-4o-mini")
# res <- interpret(df, context = "Cancer proliferation study")
print(res)

## End(Not run)
```

---

interpret_agent *Interpret enrichment results using a multi-agent pipeline (Deep Mode)*

Description

Employs three specialized AI agents in sequence for rigorous interpretation:

1. Agent Cleaner: Filters noise and selects relevant pathways.
2. Agent Detective: Identifies key regulators and functional modules.
3. Agent Synthesizer: Produces a coherent biological narrative.

Usage

```
interpret_agent(
  x,
  context = NULL,
  n_pathways = 50,
  model = NULL,
  add_ppi = FALSE,
  gene_fold_change = NULL,
  max_tokens = 8192,
  temperature = 0.3,
  verbose = FALSE
)
```

Arguments

x	An enrichment result object.
context	A string describing the experimental background.
n_pathways	Number of top pathways to consider initially. Default 50.
model	Optional LLM model. When 'NULL' (default), uses the aisdk package-wide default model configured via 'aisdk::set_model()'. You can also supply a model ID in 'provider:model' format or a 'LanguageModelV1' object. Bare model names are supported with a warning.
add_ppi	Logical, whether to query PPI data. Default FALSE.
gene_fold_change	Named numeric vector of log fold changes.
max_tokens	Maximum tokens per agent call. Default 8192.
temperature	Sampling temperature. Default 0.3.
verbose	Logical, whether to print debug messages. Default FALSE.

Details

Uses aisdk's Agent and Session system for shared context across agents.

Value

An 'interpretation' object with deep analysis fields plus regulatory_drivers, refined_network, and network_evidence from the detective agent.

Examples

```
## Not run:
res <- interpret_agent(df,
  model = "openai:gpt-4o",
  context = "scRNA-seq of mouse MI day 3"
)
print(res)

## End(Not run)
```

`interpret_hierarchical` *Interpret enrichment results using a hierarchical strategy*

Description

First interprets major clusters to establish lineage context, then interprets sub-clusters with hierarchical constraints from the major cluster annotations.

Usage

```
interpret_hierarchical(
  x_minor,
  x_major,
  mapping,
  model = NULL,
  task = "cell_type",
  max_tokens = 8192,
  temperature = 0.3
)
```

Arguments

`x_minor`	Enrichment result for sub-clusters.
`x_major`	Enrichment result for major clusters.
`mapping`	A named vector mapping sub-cluster IDs to major cluster IDs.
`model`	Optional LLM model. When ‘NULL‘ (default), uses the aisdk package-wide default model configured via ‘aisdk::set_model()‘. You can also supply a model ID in ‘provider:model‘ format or a ‘LanguageModelV1‘ object. Bare model names are supported with a warning.
`task`	Task type, default "cell_type".
`max_tokens`	Maximum tokens. Default 8192.
`temperature`	Sampling temperature. Default 0.3.

Value

An ‘interpretation_list‘ object.

ko2name *ko2name*

Description

convert ko ID to descriptive name

Usage

```
ko2name(ko)
```

Arguments

ko	ko ID

Value

data.frame

Author(s)

guangchuang yu

merge_result *merge_result*

Description

merge a list of enrichResult objects to compareClusterResult

Usage

```
merge_result(enrichResultList)
```

Arguments

enrichResultList	a list of enrichResult objects

Value

a compareClusterResult instance

Author(s)

Guangchuang Yu

plot.interpretation *plot*

Description

plot

Usage

```
## S3 method for class 'interpretation'
plot(x, layout = "nicely", ...)
```

Arguments

x	An ‘interpretation‘ object.
layout	Graph layout, default is "nicely".
...	Additional arguments passed to ‘ggplot2::ggplot‘.

plotGOgraph *plotGOgraph*

Description

plot GO graph

Usage

```
plotGOgraph(
  x,
  firstSigNodes = 10,
  useInfo = "all",
  sigForAll = TRUE,
  useFullNames = TRUE,
  ...
)
```

Arguments

x	output of enrichGO or gseGO
firstSigNodes	number of significant nodes (retangle nodes in the graph)
useInfo	additional info
sigForAll	if TRUE the score/p-value of all nodes in the DAG is shown, otherwise only score will be shown
useFullNames	logical
...	additional parameter of showSigOfNodes, please refer to topGO

Value

GO DAG graph

Author(s)

Guangchuang Yu

`read.gmt.pc` *read.gmt.pc*

Description

Parse gmt file from Pathway Common

Usage

```
read.gmt.pc(gmtfile, output = "data.frame")
```

Arguments

`gmtfile`	A gmt file
`output`	one of 'data.frame' or 'GSON'

Details

This function parse gmt file downloaded from Pathway common

Value

A data.frame or A GSON object depends on the value of 'output'

`reexports` *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

dplyr `arrange()`, `filter()`, `group_by()`, `mutate()`, `n()`, `rename()`, `select()`, `slice()`, `summarise()`

enrichit `geneID()`, `geneInCategory()`, `gsfilter()`, `setReadable()`

enrichplot `cnetplot`, `dotplot()`, `emapplot()`, `goplot()`, `gseaplot()`, `heatplot()`, `ridgeplot()`

GOSemSim `buildGOmap()`, `get_organism()`, `read.blast2go()`, `read.gaf()`

gson `read.gmt()`, `read.gmt.wp()`

magrittr `%<>%`, `%>%`

search_kegg_organism *search_kegg_organism*

Description

search kegg organism, listed in https://www.genome.jp/kegg/catalog/org_list.html

Usage

```
search_kegg_organism(
  str,
  by = "scientific_name",
  ignore.case = FALSE,
  use_internal_data = TRUE
)
```

Arguments

str	string
by	one of 'kegg.code', 'scientific_name' and 'common_name'
ignore.case	TRUE or FALSE
use_internal_data	logical, use kegg_species.rda or latest online KEGG data

Value

data.frame

Author(s)

Guangchuang Yu

simplify *simplify method*

Description

simplify output from enrichGO and gseGO by removing redundancy of enriched GO terms

simplify output from compareCluster by removing redundancy of enriched GO terms

Usage

```
## S4 method for signature 'enrichResult'
simplify(
  x,
  cutoff = 0.7,
  by = "p.adjust",
  select_fun = min,
  measure = "Wang",
  semData = NULL
)

## S4 method for signature 'gseaResult'
simplify(
  x,
  cutoff = 0.7,
  by = "p.adjust",
  select_fun = min,
  measure = "Wang",
  semData = NULL
)

## S4 method for signature 'compareClusterResult'
simplify(
  x,
  cutoff = 0.7,
  by = "p.adjust",
  select_fun = min,
  measure = "Wang",
  semData = NULL
)
```

Arguments

x	output of enrichGO
cutoff	similarity cutoff
by	feature to select representative term, selected by 'select_fun' function
select_fun	function to select feature passed by 'by' parameter
measure	method to measure similarity
semData	GOSemSimDATA object

Value

updated enrichResult object

updated compareClusterResult object

Author(s)

Guangchuang Yu

Gwang-Jin Kim and Guangchuang Yu

References

issue #28 https://github.com/GuangchuangYu/clusterProfiler/issues/28

issue #162 https://github.com/GuangchuangYu/clusterProfiler/issues/162

`uniprot_get` *uniprot_get*

Description

retreve annotation data from uniprot

Usage

```
uniprot_get(taxID)
```

Arguments

`taxID`	taxonomy ID

Value

gene table data frame

Author(s)

guangchuang yu

Index

∗ **classes**
groupGOResult-class, 22
∗ **datasets**
DataSet, 8
∗ **internal**
clusterProfiler-package, 3
infer_model_id, 33
reexports, 40
∗ **manip**
compareCluster, 6
enrichGO, 11
enrichKEGG, 13
groupGO, 21
%<>% *(*reexports*)*, 40
%>% *(*reexports*)*, 40
%<>%, *40*
%>%, *40*

append_kegg_category, 4
arrange *(*reexports*)*, 40
arrange(), *40*

bitr, 5
bitr_kegg, 5
browseKEGG, 6
buildGOmap *(*reexports*)*, 40
buildGOmap(), *40*

clusterProfiler
*(*clusterProfiler-package*)*, 3
clusterProfiler-package, 3
cnetplot, *40*
cnetplot *(*reexports*)*, 40
compareCluster, 6

DataSet, 8
DE_GSE8057 *(*DataSet*)*, 8
dotplot *(*reexports*)*, 40
dotplot(), *40*
download_KEGG, 8

dropGO, 9

emapplot *(*reexports*)*, 40
emapplot(), *40*
enrichDAVID, 9
enricher, 10
enrichGO, 11
enrichKEGG, 13
enrichMKEGG, 14
enrichPC, 15
enrichWP, 16

filter *(*reexports*)*, 40
filter(), *40*

gcSample *(*DataSet*)*, 8
geneID *(*reexports*)*, 40
geneID(), *40*
geneInCategory *(*reexports*)*, 40
geneInCategory(), *40*
get_organism *(*reexports*)*, 40
get_organism(), *40*
get_wp_organisms, 19
getPPI, 16
getTaxID, 18
getTaxInfo, 18
Gff2GeneTable, 19
go2ont, 20
go2term, 20
gofilter, 21
goplot *(*reexports*)*, 40
goplot(), *40*
group_by *(*reexports*)*, 40
group_by(), *40*
groupGO, 21
groupGOResult-class, 22
GSEA, 23
gseaplot *(*reexports*)*, 40
gseaplot(), *40*
gseGO, 24

gseKEGG, 26
gseMKEGG, 27
gsePC, 28
gseWP, 29
gsfilter *(*reexports*)*, 40
gsfilter(), *40*
gson_GO, 30
gson_GO_local, 30
gson_KEGG, 31
gson_KEGG_mapper, 31
gson_WP, 32

heatplot *(*reexports*)*, 40
heatplot(), *40*

idType, 33
infer_model_id, 33
interpret, 34
interpret_agent, 35
interpret_hierarchical, 37

kegg_category *(*DataSet*)*, 8
kegg_species *(*DataSet*)*, 8
ko2name, 38

merge_result, 38
mutate *(*reexports*)*, 40
mutate(), *40*

n *(*reexports*)*, 40
n(), *40*

plot.interpretation, 39
plotGOgraph, 39

read.blast2go *(*reexports*)*, 40
read.blast2go(), *40*
read.gaf *(*reexports*)*, 40
read.gaf(), *40*
read.gmt *(*reexports*)*, 40
read.gmt(), *40*
read.gmt.pc, 40
read.gmt.wp(), *40*
reexports, 40
rename *(*reexports*)*, 40
rename(), *40*
ridgeplot *(*reexports*)*, 40
ridgeplot(), *40*

search_kegg_organism, 41

select *(*reexports*)*, 40
select(), *40*
setReadable *(*reexports*)*, 40
setReadable(), *40*
show,groupGOResult-method
*(*groupGOResult-class*)*, 22
simplify, 41
simplify,compareClusterResult-method
*(*simplify*)*, 41
simplify,enrichResult-method
*(*simplify*)*, 41
simplify,gseaResult-method *(*simplify*)*,
41
slice *(*reexports*)*, 40
slice(), *40*
summarise *(*reexports*)*, 40
summarise(), *40*

uniprot_get, 43