

Package: clusterProfiler (via r-universe)

October 23, 2026

Type Package

Title A Universal Enrichment Tool for Interpreting Omics Data

Version 4.21.4

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. It also provides high-level access to topology-aware enrichment workflows powered by 'enrichit', including single-network and multi-layer network enrichment for Gene Ontology and KEGG analyses. Furthermore, it integrates Large Language Model (LLM) capabilities to provide automated and insightful interpretation of enrichment results.

Depends R (>= 4.2.0)

Imports aisd, AnnotationDbi, dplyr, enrichit (>= 0.1.6), 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, digest, 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

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev zlib1g-dev

Repository <https://bioc.r-universe.dev>

Date/Publication 2026-10-23 11:51:04 UTC

RemoteUrl <https://github.com/bioc/clusterProfiler>

RemoteRef HEAD

RemoteSha 59499cc447ac91c25460128ce2dad11c08a4ceb4

Contents

add_go_level	3
append_kegg_category	4
bitr	5
bitr_kegg	6
browseKEGG	7
compareCluster	7
DataSet	9
download_KEGG	9
dropGO	10
enrichDAVID	10
enricher	11
enrichGO	12
enrichKEGG	14
enrichMKEGG	15
enrichPC	16
enrichWP	17
evidence_status	18
get_ppi_network	19
get_wp_organisms	19
getPPI	20
getTaxID	21
getTaxInfo	22
Gff2GeneTable	22
go2ont	23
go2term	24
gofilter	24
groupGO	25
groupGOResult-class	26
GSEA	27
gseGO	28
gseKEGG	30
gseMKEGG	31
gsePC	33

gseWP	33
gson_GO	34
gson_GO_local	34
gson_KEGG	35
gson_KEGG_mapper	36
gson_KO	37
gson_WP	37
idType	38
interpret	38
interpret_agent	41
interpret_hierarchical	43
ko2name	44
merge_result	45
mnseGO	46
mnseKEGG	47
mnseMKEGG	49
mnseWP	50
nseGO	51
nseKEGG	52
nseMKEGG	54
nseWP	55
plot.interpretation	56
plotGOgraph	56
provenance	57
read.gmt.pc	58
reduce_go_redundancy	58
reduce_redundancy	59
search_kegg_organism	60
simplify	61
uniprot_get	63
Index	64

add_go_level	<i>Add the GO level of each enriched term</i>
--------------	---

Description

'enrichGO()'/ 'gseGO()' results carry the term, its ontology and its statistics, but not how deep the term sits in the GO hierarchy. This appends that as a 'level' column (1 = ontology root), which makes it easy to keep a band of levels by filtering, e.g. 'subset(res, level >= 3 & level <= 6)'.

Usage

```
add_go_level(x)
```

Arguments

`x` an ‘enrichResult’, ‘gseaResult’ or ‘compareClusterResult’ from a GO enrichment analysis

Details

The level is taken from the ontology the term itself belongs to, so an ‘ont = "ALL"’ (or ‘compareCluster()’) result is labelled correctly term by term. Terms that GO.db does not recognise get ‘NA’.

Value

‘x’ with a ‘level’ column appended to its result table

Author(s)

Guangchuang Yu <https://yulab-smu.top>

Examples

```
## Not run:
x <- enrichGO(gene, OrgDb = "org.Hs.eg.db", ont = "BP")
x <- add_go_level(x)
subset(x, level >= 3 & level <= 6)

## End(Not run)
```

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	<i>bitr</i>
------	-------------

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	<i>bitr_kegg</i>
-----------	------------------

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, one of "Path", "Module", "ncbi-proteinid", "ncbi-geneid", "uniprot", "kegg", and "ko"
toType	output id type, one of "Path", "Module", "ncbi-proteinid", "ncbi-geneid", "uniprot", "kegg", and "ko"
organism	supported organism, can be search using search_kegg_organism function
drop	drop NA or not

Value

data.frame

Author(s)

Guangchuang Yu

Examples

```
## Not run:  
bitr_kegg(c("P31946", "P01308"), "uniprot", "ko", "hsa")  
bitr_kegg(c("K04456", "K04526"), "ko", "uniprot", "hsa")  
  
## End(Not run)
```

browseKEGG	<i>browseKEGG</i>
------------	-------------------

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	<i>Compare gene clusters functional profile</i>
----------------	---

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 <code>Entrez~group</code> or a formula of type <code>Entrez logFC ~ group</code> 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	<i>Datasets gcSample contains a sample of gene clusters.</i>
---------	--

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	<i>download_KEGG</i>
---------------	----------------------

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

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,
    evidence = NULL
)

```

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
evidence	optional character vector of GO evidence codes to keep (for example 'c("IDA", "IPI", "IMP")', or '"IEA"' for electronic annotations only). When supplied, annotations carrying any other evidence code are dropped before the analysis — useful to avoid circular reasoning (dropping 'IEP' when the gene sets come from co-expression clustering) or to keep only manually reviewed annotations (dropping 'IEA'). Because the annotation changes, so does the background, so the resulting terms are not simply a subset of the unfiltered run. Filtered runs are not cached, so they rebuild the annotation each time. 'NULL' (default) uses every annotation.

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

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'; alternatively a 'GSON' object (e.g. from 'gson_KEGG()') holding the annotation, which allows the analysis to run without contacting KEGG
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	<i>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.</i>
-------------	--

Description

As with `enrichKEGG()`, `organism` may be a species code or a GSON object, so module enrichment can be run against a locally built annotation (see `gson_KEGG()`) when KEGG is unreachable.

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'; alternatively a 'GSON' object (e.g. from 'gson_KEGG()') holding the annotation, which allows the analysis to run without contacting KEGG
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	<i>enrichPC</i>
----------	-----------------

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

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

Details

This function performs over-representation analysis using WikiPathways

Value

A `enrichResult` instance

Author(s)

Guangchuang Yu

evidence_status	<i>Evidence status of every gene named in an interpretation</i>
-----------------	---

Description

Deterministically classifies each gene symbol that an interpretation names as observed (a member gene of the enrichment result it was given) or out_of_evidence (a valid symbol that is not). No model is called. Use it to label hypotheses from [interpret_agent\(\)](#) as inferred rather than observed: unlike a second model, this check is exact and cannot also mark grounded genes as uncertain.

Usage

```
evidence_status(
  x,
  enrichment,
  OrgDb = NULL,
  symbols = NULL,
  fields = c("overview", "key_mechanisms", "regulatory_drivers", "crosstalk",
    "hypothesis", "narrative", "reasoning", "network_evidence")
)
```

Arguments

x	An interpretation or interpretation_list returned by interpret() , interpret_agent() or interpret_hierarchical() .
enrichment	The enrichResult, compareClusterResult, gseaResult or data.frame the interpretation was produced from (readable, so that member genes are symbols).
OrgDb	Optional OrgDb used to recognise out-of-evidence symbols. Without it, only tokens that look like gene symbols and appear in symbols are considered.
symbols	Optional character vector of valid gene symbols (alternative to OrgDb).
fields	Record fields to scan.

Value

A data.frame with columns cluster, field, gene and status.

get_ppi_network	<i>get_ppi_network</i>
-----------------	------------------------

Description

get_ppi_network

Usage

```
get_ppi_network(  
  taxID = 9606,  
  version = "12.0",  
  score_threshold = 400,  
  output = "data.frame"  
)
```

Arguments

taxID	NCBI taxon identifier (e.g., 9606 for Human).
version	STRING database version (default: "12.0").
score_threshold	threshold of significance to include a interaction (default: 400).
output	one of 'data.frame' or 'igraph'

Details

Download and cache the full STRING PPI background network for a specific species.

Value

a 'data.frame' or an 'igraph' object containing the background PPI network.

get_wp_organisms	<i>get_wp_organism</i>
------------------	------------------------

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

getPPI	<i>getPPI</i>
--------	---------------

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"  
)  
  
get_ppi(  
  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

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	<i>go2ont</i>
--------	---------------

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	<i>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.</i>
---------	--

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

```
## Not run:
data(gcSample)
yy <- groupGO(gcSample[[1]], 'org.Hs.eg.db', ont="BP", level=2)
head(summary(yy))
#plot(yy)

## End(Not run)
```

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

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]

<i>GSEA</i>	<i>GSEA</i>
-------------	-------------

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,  
  seed = FALSE,  
  ...  
)
```

Arguments

<code>geneList</code>	order ranked geneList
<code>exponent</code>	weight of each step
<code>minGSSize</code>	minimal size of each geneSet for analyzing
<code>maxGSSize</code>	maximal size of genes annotated for testing
<code>eps</code>	boundary for calculating the p value in multilevel mode
<code>pvalueCutoff</code>	p-value cutoff applied to both the raw p-value and the adjusted p-value (p.adjust), consistent with the historical clusterProfiler/DOSE behavior
<code>pAdjustMethod</code>	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"
<code>gson</code>	a GSON object, if not NULL, use it as annotation data.
<code>TERM2GENE</code>	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
seed	random seed for reproducibility, set to a number (or TRUE to use a fixed default seed) to make the result reproducible, or FALSE (default) to draw a random seed on each run, so results may vary between runs. The underlying permutation engine uses its own RNG seeded with this value; see <code>enrichit::gsea()</code> for details.
...	other parameters passed to <code>enrichit::gsea_gson()</code>

Value

`gseaResult` object

Author(s)

Guangchuang Yu <https://yulab-smu.top>

<code>gseGO</code>	<i>gseGO</i>
--------------------	--------------

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,
    seed = FALSE,
    ...
)

```

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", "permute" and "sample"
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
seed	random seed for reproducibility, set to a number (or TRUE to use a fixed default seed) to make the result reproducible, or FALSE (default) to draw a random seed on each run, so results may vary between runs. The underlying permutation engine uses its own RNG seeded with this value; see <code>enrichit::gsea()</code> for details.
...	other parameters passed to <code>enrichit::gsea_gson()</code>

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,
  seed = FALSE,
  ...
)

```

Arguments

geneList	order ranked geneList
organism	supported organism listed in ' https://www.genome.jp/kegg/catalog/org_list.html '; alternatively a 'GSON' object (e.g. from 'gson_KEGG()') holding the annotation, which allows the analysis to run without contacting KEGG
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", "permute" and "sample"
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
seed	random seed for reproducibility, set to a number (or TRUE to use a fixed default seed) to make the result reproducible, or FALSE (default) to draw a random seed on each run, so results may vary between runs. The underlying permutation engine uses its own RNG seeded with this value; see <code>enrichit::gsea()</code> for details.
...	other parameters passed to <code>enrichit::gsea_gson()</code>

Value

`gseaResult` object

Author(s)

Yu Guangchuang

<code>gseMKEGG</code>	<i>gseMKEGG</i>
-----------------------	-----------------

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,
    seed = FALSE,
    ...
)

```

Arguments

geneList	order ranked geneList
organism	supported organism listed in ' https://www.genome.jp/kegg/catalog/org_list.html '; alternatively a 'GSON' object (e.g. from 'gson_KEGG()') holding the annotation, which allows the analysis to run without contacting KEGG
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", "permute" and "sample"
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
seed	random seed for reproducibility, set to a number (or TRUE to use a fixed default seed) to make the result reproducible, or FALSE (default) to draw a random seed on each run, so results may vary between runs. The underlying permutation engine uses its own RNG seeded with this value; see <code>enrichit::gsea()</code> for details.
...	other parameters passed to <code>enrichit::gsea_gson()</code>

Value

gseaResult object

Author(s)

Yu Guangchuang

*gsePC**gsePC*

Description

GSEA analysis for Pathway Commons

Usage

```
gsePC(geneList, eps = 1e-10, ...)
```

Arguments

<code>geneList</code>	a ranked gene list
<code>eps</code>	boundary for calculating the p value in multilevel mode
<code>...</code>	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, eps = 1e-10, ...)
```

Arguments

<code>geneList</code>	ranked gene list
<code>organism</code>	supported organisms, which can be accessed via the <code>get_wp_organisms()</code> function
<code>eps</code>	boundary for calculating the p value in multilevel mode
<code>...</code>	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

<code>gson_GO</code>	<i><code>gson_KEGG</code></i>
----------------------	-------------------------------

Description

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

Usage

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

Arguments

<code>OrgDb</code>	<code>OrgDb</code>
<code>keytype</code>	keytype of genes.
<code>ont</code>	one of "BP", "MF", "CC", and "ALL"

Value

a 'GSON' object

<code>gson_GO_local</code>	<i>Build a gson object that annotate Gene Ontology</i>
----------------------------	--

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	<i>gson_KEGG</i>
-----------	------------------

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', 'uniprot', and 'ko'.

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_KO**gson_KO*

Description

Download KEGG Orthology annotation as a GSON object

Usage

```
gson_KO()
```

Value

a 'GSON' object

*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 <code>get_wp_organisms()</code> function.
----------	---

idType	<i>idType</i>
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	
interpret	<i>Interpret Enrichment Results Using LLMs</i>

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 = NULL, model = NULL, add_ppi = FALSE, gene_fold_change = NULL, gene_stats = NULL,

```

    max_tokens = 8192,
    temperature = 0.3,
    reduce = TRUE,
    reduce_threshold = NULL,
    reasoning = NULL,
    verbose = FALSE,
    reduce_go = NULL,
    go_reduction_threshold = NULL
  )

```

Arguments

<code>x</code>	An enrichment result object (<code>enrichResult</code> , <code>gseaResult</code> , <code>compareClusterResult</code> , or a <code>data.frame</code> with pathway columns).
<code>context</code>	A string describing the experimental background (e.g., "scRNA-seq of mouse myocardial infarction at day 3").
<code>n_pathways</code>	Maximum number of terms (per cluster) sent to the model, taken in order of adjusted p-value after reduction. <code>NULL</code> (default) sends every term that passes the object's cutoffs; with <code>reduce = TRUE</code> that is one representative per theme. On five public lists a cap below the number of representatives named fewer themes of the full result; for a model with a short context, lower <code>reduce_threshold</code> (for example 0.3) before setting a cap.
<code>model</code>	Optional LLM model. When <code>NULL</code> (default), uses the <code>aisdk</code> package-wide default model configured via <code>aisdk::set_model()</code> . You can also supply a model ID in <code>provider:model</code> format (e.g., "deepseek:deepseek-chat", "gemini:gemini-2.5-flash") or a <code>LanguageModelV1</code> object. Bare model names are supported with a warning (e.g., "deepseek-chat").
<code>add_ppi</code>	Logical, whether to query STRING PPI network data. Default <code>FALSE</code> .
<code>gene_fold_change</code>	Named numeric vector of log fold changes for expression context. For a <code>gseaResult</code> left <code>NULL</code> , the ranked list the GSEA was run on (<code>x@geneList</code>) is used, so the leading-edge genes reach the model with their values.
<code>gene_stats</code>	Optional data frame of per-gene statistics from the differential analysis behind <code>x</code> (a <code>limma topTable()</code> , <code>DESeq2 results()</code> , <code>edgeR topTags()</code> or <code>Seurat FindMarkers()</code> table). Recognised columns: the gene (<code>gene</code> , <code>SYMBOL</code> , <code>symbol</code> , <code>gene_symbol</code> , <code>ENTREZID</code> , <code>ID</code> , or the row names), the log fold change (<code>log2FC</code> , <code>logFC</code> , <code>log2FoldChange</code> , <code>avg_log2FC</code>) and the adjusted p-value (<code>padj</code> , <code>adj.P.Val</code> , <code>FDR</code> , <code>p_val_adj</code> , <code>p.adjust</code>). When supplied, the genes of the enriched terms are listed with log fold change and FDR, and the most significant up- and down-regulated genes are added, so that the genes the model names are read from the data rather than from the term names. It also supplies <code>gene_fold_change</code> when that is <code>NULL</code> .
<code>max_tokens</code>	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.
<code>temperature</code>	Sampling temperature. Default 0.3.

reduce	Logical. Run <code>reduce_redundancy()</code> before building the prompt, so every significant theme reaches the model once: Wang similarity for GO results, Jaccard overlap of member genes for KEGG and other gene-set collections. Default TRUE; a no-op for data frames and lists.
reduce_threshold	Similarity threshold passed to <code>reduce_redundancy()</code> . NULL (default) uses its default of 0.5.
reasoning	Optional reasoning/thinking configuration passed through to the provider, e.g. <code>list(type = "enabled", budget_tokens = 4096)</code> . Left NULL the request is byte-identical to one made before this argument existed. Whatever is passed is recorded in the execution stamp as <code>reasoning_requested</code> , alongside <code>reasoning_evidence</code> – whether the response actually carried a reasoning block.
verbose	Logical, whether to print debug messages showing raw API responses, token usage, and JSON parsing details. Default FALSE. Equivalent to setting <code>options(aisdk.debug = TRUE)</code> for the call.
reduce_go, go_reduction_threshold	Earlier names of <code>reduce</code> and <code>reduce_threshold</code> , still accepted; a call that passes <code>reduce_go</code> keeps its earlier default threshold of 0.7.

Details

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

Value

An interpretation object (list): `overview`, `regulatory_drivers`, `key_mechanisms`, `crosstalk`, `hypothesis`, `narrative`, `refined_network` and `network_evidence`. `interpret()` interprets the enrichment result and nothing else; to combine it with the results of other tools (a SingleR label, a TF-activity ranking, a complex enrichment) and reach a conclusion, use `interpret_agent()` with prior.

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"
)
# Every significant term, one representative per theme (the default)
ego <- enrichGO(gene = genes, OrgDb = org.Mm.eg.db, ont = "BP")
res <- interpret(ego)
# A shorter prompt for a model with a short context: merge more terms per theme
```

```
res <- interpret(ego, reduce_threshold = 0.3)
# 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 specialized AI agents in sequence for rigorous interpretation, and is the entry point that combines the enrichment evidence with the results of other tools (prior) into a conclusion:

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.
4. Agent Reviewer (optional, review = TRUE): Audits the draft for grounding, overclaiming, and experimental-label fidelity; flags issues and returns a corrected overview.

Usage

```
interpret_agent(
  x,
  context = NULL,
  prior = NULL,
  n_pathways = NULL,
  model = NULL,
  add_ppi = FALSE,
  gene_fold_change = NULL,
  max_tokens = 8192,
  temperature = 0.3,
  review = FALSE,
  reduce = TRUE,
  reduce_threshold = NULL,
  reasoning = NULL,
  verbose = FALSE,
  reduce_go = NULL,
  go_reduction_threshold = NULL
)
```

Arguments

x	An enrichment result object.
context	A string describing the experimental background. State the question the analysis should answer here (for example "which cell type is this cluster?"); the answer is returned as conclusion.
prior	Optional results of other tools to be checked against the enrichment evidence, as text: a SingleR or scGPT label, a ranked list of transcription factors from a regulon analysis, the protein complexes enriched in the gene list, and so on. A single string applies to every cluster; a named character vector or list supplies one entry per cluster. The Detective and the Synthesizer see it as a claim to check, not as evidence, and the result gains <code>prior_verdict</code> (Confirmed, Refined, Corrected or Not assessable) and <code>prior_rationale</code> .
n_pathways	Maximum number of terms (per cluster) the Cleaner starts from, taken in order of adjusted p-value after reduction. NULL (default) uses every term that passes the object's cutoffs; with <code>reduce = TRUE</code> that is one representative per theme.
model	Optional LLM model. When NULL (default), uses the <code>aisdk</code> package-wide default model configured via <code>aisdk::set_model()</code> . You can also supply a model ID in <code>provider:model</code> format or a <code>LanguageModelV1</code> 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. When supplied, every input gene is also listed to the Detective and the Synthesizer, most up-regulated first.
max_tokens	Maximum tokens per agent call. Default 8192.
temperature	Sampling temperature. Default 0.3.
review	Logical, whether to run a final Reviewer agent (Agent 4) that audits the synthesized draft against the input evidence for grounding, overclaiming, experimental-label fidelity, and evidence sufficiency (whether the supplied evidence classes can support a claim as specific as the draft makes). When TRUE, the result gains a <code>review</code> field (<code>verdict</code> , <code>grounding_score</code> , <code>evidence_sufficiency</code> , <code>supported_resolution</code> , <code>abstain_recommended</code> , <code>issues</code> , <code>adjusted_confidence</code> , <code>warnings</code>) and <code>overview_original</code> ; if the verdict is "revise", <code>overview</code> is replaced by the corrected version, in which claims exceeding <code>supported_resolution</code> are downgraded to that resolution. Default FALSE, which reproduces the released three-agent behavior exactly: the audit is an addition a caller opts into, and leaving it off keeps existing scripts measuring what they measured before. Note that the audit roughly doubles wall-clock time per unit and adds one model call, and that it is not infallible – on objects where the Synthesizer returned nothing it has been observed to return pass with a High grounding score.
reduce	Logical. Run <code>reduce_redundancy()</code> before building any prompt, as <code>interpret()</code> does: Wang similarity for GO results, Jaccard overlap of member genes for other gene-set collections. Deterministic and makes no model call. Default TRUE.
reduce_threshold	Similarity threshold passed to <code>reduce_redundancy()</code> . NULL (default) uses its default of 0.5.

reasoning	Optional reasoning/thinking configuration passed through to the provider, e.g. <code>list(type = "enabled", budget_tokens = 4096)</code> . Left NULL the request is byte-identical to one made before this argument existed. Whatever is passed is recorded in the execution stamp as <code>reasoning_requested</code> , alongside <code>reasoning_evidence</code> – whether the response actually carried a reasoning block. The two are separate because a gateway can accept the parameter and drop it, and the caller can only observe the second.
verbose	Logical, whether to print debug messages. Default FALSE.
reduce_go, go_reduction_threshold	Earlier names of <code>reduce</code> and <code>reduce_threshold</code> , still accepted; a call that passes <code>reduce_go</code> keeps its earlier default threshold of 0.7.

Details

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

Value

An interpretation object with deep analysis fields, `conclusion` (the answer to the question the context poses), plus `regulatory_drivers`, `refined_network`, and `network_evidence` from the detective agent. When prior is supplied, also `prior_verdict` and `prior_rationale`. When `review = TRUE`, also includes `review` and `overview_original`. The `review` field reports grounding separately from resolution: `abstain_recommended = "yes"` flags a draft whose cited evidence is real but which resolves to a more specific entity than that evidence can single out, with `supported_resolution` giving the claim the evidence does support.

Examples

```
## Not run:
res <- interpret_agent(df,
  model = "openai:gpt-4o",
  context = "scRNA-seq of mouse MI day 3"
)
print(res)
# Check another tool's answer against the enrichment evidence
res <- interpret_agent(ego_cluster3,
  context = "PBMC scRNA-seq; which cell type is cluster 3?",
  gene_fold_change = fc3, prior = "SingleR (HPCA): T_cell:CD4+_Naive"
)
res$conclusion; res$prior_verdict

## 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,  
  max_tokens = 8192,  
  temperature = 0.3,  
  n_pathways = NULL,  
  reduce = TRUE,  
  reduce_threshold = NULL  
)
```

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.
max_tokens	Maximum tokens. Default 8192.
temperature	Sampling temperature. Default 0.3.
n_pathways, reduce, reduce_threshold	Passed to both levels, as in interpret(): by default every significant term, one representative per theme.

Value

An 'interpretation_list' object with a cell-type label ('cell_type', 'confidence', 'reasoning', 'markers') per sub-cluster.

ko2name	ko2name
---------	---------

Description

convert ko ID to descriptive name

Usage

```
ko2name(ko)
```

Arguments

```
ko          ko ID
```

Value

```
data.frame
```

Author(s)

```
guangchuang yu
```

<code>merge_result</code>	<i>merge_result</i>
---------------------------	---------------------

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
```

mnseGO

*mnseGO***Description**

Multi-layer Network-based Gene Set Enrichment Analysis of Gene Ontology

Usage

```
mnseGO(
  seed_list,
  networks,
  couplings,
  ont = "BP",
  OrgDb,
  keyType = "ENTREZID",
  mode = c("evidence", "signed"),
  layer_weights = NULL,
  collapse = c("weighted_mean", "sum", "mean", "max_abs"),
  target_layer = NULL,
  output_space = c("union", "gene"),
  p = 0.5,
  interlayer_strength = 1,
  specific_weight = FALSE,
  minGSSize = 10,
  maxGSSize = 500,
  threshold = 1e-09,
  maxIter = 100,
  verbose = TRUE,
  ...
)
```

Arguments

<code>seed_list</code>	named list of named numeric vectors, one per layer
<code>networks</code>	named list of layer-specific networks
<code>couplings</code>	data.frame of inter-layer edges
<code>ont</code>	one of "BP", "MF", and "CC" subontologies, or "ALL" for all three.
<code>OrgDb</code>	OrgDb
<code>keyType</code>	keytype of gene
<code>mode</code>	propagation mode, one of "evidence" and "signed"
<code>layer_weights</code>	optional named numeric vector of layer weights
<code>collapse</code>	one of "weighted_mean", "sum", "mean", or "max_abs"
<code>target_layer</code>	optional layer name to export scores from

output_space one of "union" and "gene"
 p restart probability for RWR
 interlayer_strength global scaling factor for coupling edges
 specific_weight logical, whether to apply gene specificity weighting
 minGSSize minimal size of each geneSet for analyzing
 maxGSSize maximal size of genes annotated for testing
 threshold convergence threshold for RWR
 maxIter maximal number of RWR iterations
 verbose print message or not
 ... other parameters passed to `enrichit::mnsea_gson()`

Value

mnseaResult object

mnseKEGG

mnseKEGG

Description

Multi-layer Network-based Gene Set Enrichment Analysis of KEGG

Usage

```

mnseKEGG(
  seed_list,
  networks,
  couplings,
  organism = "hsa",
  keyType = "kegg",
  mode = c("evidence", "signed"),
  layer_weights = NULL,
  collapse = c("weighted_mean", "sum", "mean", "max_abs"),
  target_layer = NULL,
  output_space = c("union", "gene"),
  p = 0.5,
  interlayer_strength = 1,
  specific_weight = FALSE,
  minGSSize = 10,
  maxGSSize = 500,
  threshold = 1e-09,
  maxIter = 100,
  verbose = TRUE,

```

```

    use_internal_data = FALSE,
    ...
)

```

Arguments

<code>seed_list</code>	named list of named numeric vectors, one per layer
<code>networks</code>	named list of layer-specific networks
<code>couplings</code>	data.frame of inter-layer edges
<code>organism</code>	supported organism listed in ' https://www.genome.jp/kegg/catalog/org_list.html ', or a GSON object
<code>keyType</code>	one of "kegg", 'ncbi-geneid', 'ncbi-proteinid' and 'uniprot'
<code>mode</code>	propagation mode, one of "evidence" and "signed"
<code>layer_weights</code>	optional named numeric vector of layer weights
<code>collapse</code>	one of "weighted_mean", "sum", "mean", or "max_abs"
<code>target_layer</code>	optional layer name to export scores from
<code>output_space</code>	one of "union" and "gene"
<code>p</code>	restart probability for RWR
<code>interlayer_strength</code>	global scaling factor for coupling edges
<code>specific_weight</code>	logical, whether to apply gene specificity weighting
<code>minGSSize</code>	minimal size of each geneSet for analyzing
<code>maxGSSize</code>	maximal size of genes annotated for testing
<code>threshold</code>	convergence threshold for RWR
<code>maxIter</code>	maximal number of RWR iterations
<code>verbose</code>	print message or not
<code>use_internal_data</code>	logical, use KEGG.db or latest online KEGG data
<code>...</code>	other parameters passed to <code>enrichit::mnsea_gson()</code>

Value

mnseaResult object

mnseMKEGG

*mnseMKEGG***Description**

Multi-layer Network-based Gene Set Enrichment Analysis of KEGG Module

Usage

```
mnseMKEGG(
  seed_list,
  networks,
  couplings,
  organism = "hsa",
  keyType = "kegg",
  mode = c("evidence", "signed"),
  layer_weights = NULL,
  collapse = c("weighted_mean", "sum", "mean", "max_abs"),
  target_layer = NULL,
  output_space = c("union", "gene"),
  p = 0.5,
  interlayer_strength = 1,
  specific_weight = FALSE,
  minGSSize = 10,
  maxGSSize = 500,
  threshold = 1e-09,
  maxIter = 100,
  verbose = TRUE,
  ...
)
```

Arguments

seed_list	named list of named numeric vectors, one per layer
networks	named list of layer-specific networks
couplings	data.frame of inter-layer edges
organism	supported organism listed in ' https://www.genome.jp/kegg/catalog/org_list.html ', or a GSON object
keyType	one of "kegg", 'ncbi-geneid', 'ncbi-proteinid' and 'uniprot'
mode	propagation mode, one of "evidence" and "signed"
layer_weights	optional named numeric vector of layer weights
collapse	one of "weighted_mean", "sum", "mean", or "max_abs"
target_layer	optional layer name to export scores from
output_space	one of "union" and "gene"

p restart probability for RWR
interlayer_strength global scaling factor for coupling edges
specific_weight logical, whether to apply gene specificity weighting
minGSSize minimal size of each geneSet for analyzing
maxGSSize maximal size of genes annotated for testing
threshold convergence threshold for RWR
maxIter maximal number of RWR iterations
verbose print message or not
... other parameters passed to enrichit::mnsea_gson()

Value

mnseaResult object

mnseWP	<i>mnseWP</i>
--------	---------------

Description

Multi-layer Network-based Gene Set Enrichment Analysis of WikiPathways

Usage

```
mnseWP(  
  seed_list,  
  networks,  
  couplings,  
  organism,  
  mode = c("evidence", "signed"),  
  layer_weights = NULL,  
  collapse = c("weighted_mean", "sum", "mean", "max_abs"),  
  target_layer = NULL,  
  output_space = c("union", "gene"),  
  p = 0.5,  
  interlayer_strength = 1,  
  specific_weight = FALSE,  
  minGSSize = 10,  
  maxGSSize = 500,  
  threshold = 1e-09,  
  maxIter = 100,  
  verbose = TRUE,  
  ...  
)
```

Arguments

seed_list	named list of named numeric vectors, one per layer
networks	named list of layer-specific networks
couplings	data.frame of inter-layer edges
organism	supported organisms listed by <code>get_wp_organisms()</code> , or a GSON object
mode	propagation mode, one of "evidence" and "signed"
layer_weights	optional named numeric vector of layer weights
collapse	one of "weighted_mean", "sum", "mean", or "max_abs"
target_layer	optional layer name to export scores from
output_space	one of "union" and "gene"
p	restart probability for RWR
interlayer_strength	global scaling factor for coupling edges
specific_weight	logical, whether to apply gene specificity weighting
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of genes annotated for testing
threshold	convergence threshold for RWR
maxIter	maximal number of RWR iterations
verbose	print message or not
...	other parameters passed to <code>enrichit::mnsea_gson()</code>

Value

mnseaResult object

nseGO

nseGO

Description

Network-based Gene Set Enrichment Analysis of Gene Ontology

Usage

```
nseGO(
  geneList,
  network,
  ont = "BP",
  OrgDb,
  keyType = "ENTREZID",
  mode = c("evidence", "signed"),
```

```
    p = 0.5,
    specific_weight = FALSE,
    minGSSize = 10,
    maxGSSize = 500,
    threshold = 1e-09,
    maxIter = 100,
    verbose = TRUE,
    ...
  )
```

Arguments

geneList	order ranked geneList
network	edge list (data.frame with 2 or 3 columns) or sparse matrix
ont	one of "BP", "MF", and "CC" subontologies, or "ALL" for all three.
OrgDb	OrgDb
keyType	keytype of gene
mode	propagation mode, one of "evidence" and "signed"
p	restart probability for RWR
specific_weight	logical, whether to apply gene specificity weighting
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of genes annotated for testing
threshold	convergence threshold for RWR
maxIter	maximal number of RWR iterations
verbose	print message or not
...	other parameters passed to enrichit::nsea_gson()

Value

nseaResult object

nseKEGG	<i>nseKEGG</i>
---------	----------------

Description

Network-based Gene Set Enrichment Analysis of KEGG

Usage

```
nseKEGG(
  geneList,
  network,
  organism = "hsa",
  keyType = "kegg",
  mode = c("evidence", "signed"),
  p = 0.5,
  specific_weight = FALSE,
  minGSSize = 10,
  maxGSSize = 500,
  threshold = 1e-09,
  maxIter = 100,
  verbose = TRUE,
  use_internal_data = FALSE,
  ...
)
```

Arguments

geneList	order ranked geneList
network	edge list (data.frame with 2 or 3 columns) or sparse matrix
organism	supported organism listed in ' https://www.genome.jp/kegg/catalog/org_list.html ', or a GSON object
keyType	one of "kegg", 'ncbi-geneid', 'ncbi-proteinid' and 'uniprot'
mode	propagation mode, one of "evidence" and "signed"
p	restart probability for RWR
specific_weight	logical, whether to apply gene specificity weighting
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of genes annotated for testing
threshold	convergence threshold for RWR
maxIter	maximal number of RWR iterations
verbose	print message or not
use_internal_data	logical, use KEGG.db or latest online KEGG data
...	other parameters passed to <code>enrichit::nsea_gson()</code>

Value

nseaResult object

nseMKEGG

*nseMKEGG***Description**

Network-based Gene Set Enrichment Analysis of KEGG Module

Usage

```
nseMKEGG(
  geneList,
  network,
  organism = "hsa",
  keyType = "kegg",
  mode = c("evidence", "signed"),
  p = 0.5,
  specific_weight = FALSE,
  minGSSize = 10,
  maxGSSize = 500,
  threshold = 1e-09,
  maxIter = 100,
  verbose = TRUE,
  ...
)
```

Arguments

geneList	order ranked geneList
network	edge list (data.frame with 2 or 3 columns) or sparse matrix
organism	supported organism listed in ' https://www.genome.jp/kegg/catalog/org_list.html ', or a GSON object
keyType	one of "kegg", 'ncbi-geneid', 'ncbi-proteinid' and 'uniprot'
mode	propagation mode, one of "evidence" and "signed"
p	restart probability for RWR
specific_weight	logical, whether to apply gene specificity weighting
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of genes annotated for testing
threshold	convergence threshold for RWR
maxIter	maximal number of RWR iterations
verbose	print message or not
...	other parameters passed to <code>enrichit::nsea_gson()</code>

Value

nseaResult object

nseWP	<i>nseWP</i>
-------	--------------

Description

Network-based Gene Set Enrichment Analysis of WikiPathways

Usage

```
nseWP(  
  geneList,  
  network,  
  organism,  
  mode = c("evidence", "signed"),  
  p = 0.5,  
  specific_weight = FALSE,  
  minGSSize = 10,  
  maxGSSize = 500,  
  threshold = 1e-09,  
  maxIter = 100,  
  verbose = TRUE,  
  ...  
)
```

Arguments

geneList	order ranked geneList
network	edge list (data.frame with 2 or 3 columns) or sparse matrix
organism	supported organisms listed by <code>get_wp_organisms()</code> , or a GSON object
mode	propagation mode, one of "evidence" and "signed"
p	restart probability for RWR
specific_weight	logical, whether to apply gene specificity weighting
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of genes annotated for testing
threshold	convergence threshold for RWR
maxIter	maximal number of RWR iterations
verbose	print message or not
...	other parameters passed to <code>enrichit::nsea_gson()</code>

Value

nseaResult object

plot.interpretation	<i>plot</i>
---------------------	-------------

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	<i>plotGOgraph</i>
-------------	--------------------

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 (rectangle 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

provenance	<i>Execution provenance of an interpretation</i>
------------	--

Description

Returns the execution record attached to a result of `interpret()`, `interpret_agent()` or `interpret_hierarchical()`: the model actually addressed, the decoding parameters requested (and whether the response confirmed them), SHA-256 digests of the system prompt, user prompt, raw response and input evidence, token usage, finish_reason, per-call status and package versions.

Usage

`provenance(x)`

Arguments

`x` An interpretation, `interpretation_list` or agent result.

Details

A remote model is not a versionable dependency. Without this record an interpretation cannot be reproduced, and a call that failed at the transport layer is indistinguishable from a model that had nothing to say – both arrive as an ordinary result object. Check `provenance(x)$ok` before treating any interpretation as data.

Value

An `interpret_provenance` list, or NULL if the object carries no stamp.

read.gmt.pc	<i>read.gmt.pc</i>
-------------	--------------------

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'

reduce_go_redundancy	<i>Reduce GO term redundancy by semantic similarity</i>
----------------------	---

Description

Applies semantic similarity-based clustering to remove redundant GO terms before LLM interpretation. This algorithmic preprocessing complements the multi-agent interpretation layer by reducing input noise and token count.

Usage

```
reduce_go_redundancy(  
  enrich_result,  
  threshold = 0.7,  
  orgdb = NULL,  
  method = "Wang"  
)
```

Arguments

enrich_result	An enrichResult object from enrichGO()
threshold	Semantic similarity threshold (0-1). Higher values are more aggressive. Default 0.7 means terms with >70% similarity are clustered.
orgdb	OrgDb package name (e.g., "org.Hs.eg.db"). If NULL, auto-detected from the enrichResult organism field.
method	Similarity measure passed to <code>GOSemSim::goSim()</code> . Default "Wang", which reads the GO DAG directly and needs no corpus statistics. The information-content measures "Resnik", "Lin", "Rel" and "Jiang" are accepted but require <code>godata(computeIC = TRUE)</code> and are substantially slower.

Details

Implemented directly on `GOSemSim`, which is already an Imports dependency. The `rrvgo_*` attribute names are kept for backwards compatibility with existing ablation outputs and predate the switch away from `rrvgo`.

Value

A reduced `enrichResult` object with redundant terms removed. The object carries attributes recording the reduction: `rrvgo_reduced`, `rrvgo_original_n`, `rrvgo_reduced_n`, `rrvgo_threshold`, `rrvgo_method`.

Examples

```
## Not run:
ego <- enrichGO(gene = genes, OrgDb = org.Mm.eg.db, ont = "BP")
ego_reduced <- reduce_go_redundancy(ego, threshold = 0.7)
message(sprintf("Reduced from %d to %d terms",
  attr(ego_reduced, "rrvgo_original_n"),
  attr(ego_reduced, "rrvgo_reduced_n")))

## End(Not run)
```

reduce_redundancy	<i>Reduce redundant terms in an enrichment result before interpretation</i>
-------------------	---

Description

Keeps one representative for each group of near-duplicate significant terms, so that every significant theme of a long result can be shown to the model without repeating its near-synonyms. GO results are grouped by Wang semantic similarity over the GO DAG (`GOSemSim`); every other gene-set collection (KEGG, Reactome, WikiPathways, MSigDB, custom TERM2GENE) by the Jaccard overlap of the member genes the result reports (`geneID`, or `core_enrichment` for GSEA). Groups are formed greedily in order of adjusted p-value: the most significant remaining term is kept and every remaining term more similar to it than threshold is dropped. Only terms that pass the object's cutoffs are considered; a `compareClusterResult` is reduced within each cluster, and a GO result spanning

several ontologies within each ontology. Deterministic, and makes no model call. `interpret()` and `interpret_agent()` call it by default (reduce = TRUE).

Usage

```
reduce_redundancy(
  x,
  threshold = NULL,
  method = c("auto", "Wang", "jaccard"),
  semData = NULL
)
```

Arguments

<code>x</code>	An <code>enrichResult</code> , <code>gseaResult</code> or <code>compareClusterResult</code> .
<code>threshold</code>	Similarity above which a term joins the group of a more significant term. NULL (default) uses 0.5 for both measures.
<code>method</code>	"auto" (default) uses "Wang" when every term is a GO term and "jaccard" otherwise; either can be forced. Wang on non-GO terms falls back to Jaccard.
<code>semData</code>	Optional <code>GOSemSimDATA</code> for the Wang measure; built with <code>GOSemSim::godata(ont =)</code> when NULL. Wang reads only the GO DAG, so no <code>OrgDb</code> is needed.

Value

`x` with the redundant significant terms removed. Attributes record the step: `reduced`, `n_original`, `n_reduced`, `reduction_method` and `reduction_threshold`, and the legacy `rrvgo_*` names read by earlier analysis scripts.

Examples

```
## Not run:
ego <- enrichGO(gene = genes, OrgDb = org.Hs.eg.db, ont = "BP")
reduce_redundancy(ego)           # Wang, 0.5
kk <- enrichKEGG(gene = genes, organism = "hsa")
reduce_redundancy(kk)           # Jaccard on member genes, 0.5

## End(Not run)
```

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	<i>simplify method</i>
----------	------------------------

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	<i>uniprot_get</i>
-------------	--------------------

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, [26](#)
- * **datasets**
 - DataSet, [9](#)
- * **manip**
 - compareCluster, [7](#)
 - enrichGO, [12](#)
 - enrichKEGG, [14](#)
 - groupGO, [25](#)
- add_go_level, [3](#)
- append_kegg_category, [4](#)
- bitr, [5](#)
- bitr_kegg, [6](#)
- browseKEGG, [7](#)
- compareCluster, [7](#)
- DataSet, [9](#)
- DE_GSE8057 (DataSet), [9](#)
- download_KEGG, [9](#)
- dropGO, [10](#)
- enrichDAVID, [10](#)
- enricher, [11](#)
- enrichGO, [12](#)
- enrichKEGG, [14](#)
- enrichMKEGG, [15](#)
- enrichPC, [16](#)
- enrichWP, [17](#)
- evidence_status, [18](#)
- gcSample (DataSet), [9](#)
- get_ppi (getPPI), [20](#)
- get_ppi_network, [19](#)
- get_wp_organisms, [19](#)
- getPPI, [20](#)
- getTaxID, [21](#)
- getTaxInfo, [22](#)
- Gff2GeneTable, [22](#)
- go2ont, [23](#)
- go2term, [24](#)
- gofilter, [24](#)
- GOSemSim::goSim(), [59](#)
- groupGO, [25](#)
- groupGOResult-class, [26](#)
- GSEA, [27](#)
- gseGO, [28](#)
- gseKEGG, [30](#)
- gseMKEGG, [31](#)
- gsePC, [33](#)
- gseWP, [33](#)
- gson_GO, [34](#)
- gson_GO_local, [34](#)
- gson_KEGG, [35](#)
- gson_KEGG_mapper, [36](#)
- gson_KO, [37](#)
- gson_WP, [37](#)
- idType, [38](#)
- interpret, [38](#)
- interpret(), [18](#), [42](#), [57](#), [60](#)
- interpret_agent, [41](#)
- interpret_agent(), [18](#), [40](#), [57](#), [60](#)
- interpret_hierarchical, [43](#)
- interpret_hierarchical(), [18](#), [57](#)
- kegg_category (DataSet), [9](#)
- kegg_species (DataSet), [9](#)
- ko2name, [44](#)
- merge_result, [45](#)
- mnseGO, [46](#)
- mnseKEGG, [47](#)
- mnseMKEGG, [49](#)
- mnseWP, [50](#)
- nseGO, [51](#)
- nseKEGG, [52](#)
- nseMKEGG, [54](#)

nseWP, [55](#)

plot.interpretation, [56](#)

plotGOgraph, [56](#)

provenance, [57](#)

read.gmt.pc, [58](#)

reduce_go_redundancy, [58](#)

reduce_redundancy, [59](#)

reduce_redundancy(), [40](#), [42](#)

search_kegg_organism, [60](#)

show, groupGOResult-method
 (groupGOResult-class), [26](#)

simplify, [61](#)

simplify, compareClusterResult-method
 (simplify), [61](#)

simplify, enrichResult-method
 (simplify), [61](#)

simplify, gseaResult-method (simplify),
 [61](#)

uniprot_get, [63](#)