

Package ‘enrichit’

September 17, 2026

Title 'C++' Implementations of Functional Enrichment Analysis

Version 0.2.4

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Description Fast implementations of functional enrichment analysis methods using 'C++' via 'Rcpp'.

Currently provides Over-

Representation Analysis (ORA), Gene Set Enrichment Analysis (GSEA),

Weighted Enrichment Analysis for ORA and GSEA, Network-

based Set Enrichment Analysis (NSEA),

multi-layer network-based enrichment, and multi-omics integration workflows. Additional

features include early fusion at the feature level, late fusion at the pathway level,

multi-omics contribution tracing, topology-aware explanation helpers, Bayesian term

selection, and extremely fast Random Walk with Restart (RWR) using 'RcppEigen'. The

enrichment methods build on GSEA by Subramanian et al. (2005)

<[doi:10.1073/pnas.0506580102](https://doi.org/10.1073/pnas.0506580102)>, the multilevel strategy derived from 'fgsea'

by Korotkevich et al. (2021) <[doi:10.1101/060012](https://doi.org/10.1101/060012)>, and network-based

enrichment ideas described by Glaab et al. (2012)

<[doi:10.1093/bioinformatics/bts389](https://doi.org/10.1093/bioinformatics/bts389)>.

License Artistic-2.0

Depends R (>= 3.5.0)

Imports Matrix, methods, Rcpp (>= 1.0.10), rlang, stats, yulab.utils
(> 0.2.1)

LinkingTo Rcpp, RcppEigen

Suggests AnnotationDbi, BiasedUrn, clusterProfiler, DOSE, fgsea, gson,
qvalue, testthat

Encoding UTF-8

URL <https://yulab-smu.top/biomedical-knowledge-mining-book/>

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Guangchuang Yu [aut, cre]

Repository CRAN

Date/Publication 2026-09-17 09:10:08 UTC

Contents

aggregate_enrichment	2
aggregate_omics	3
bayes_enrich	4
bayes_summary	5
classify_omics_pattern	6
collapse_multilayer_scores	6
compareClusterResult-class	7
enrichResult-class	8
EXTID2NAME	9
extract_mnsea_subnetwork	9
geneID	10
geneInCategory	11
get_mnsea_contribution	11
get_omics_contribution	12
gsea	12
gseaResult-class	14
gseaScores	15
gsea_gson	16
gsfilter	17
harmonize_ids	18
mnsea	18
mnseaResult-class	20
mnsea_gson	21
nsea	22
nseaResult-class	24
nsea_gson	25
ora	26
ora_gson	27
prepare_multilayer_network	28
prepare_network	29
propagate_multilayer	30
select_features_for_ora	31
setReadable	31
show	32
summary	33
Index	34

aggregate_enrichment *Aggregate multiple enrichment results (Late Fusion)*

Description

Combine pathway-level enrichment results from multiple omics or independent analyses. P-values of identical pathways are merged using statistical methods (e.g., Brown's method).

Usage

```
aggregate_enrichment(res_list, method = c("brown", "fisher", "stouffer"), ...)
```

Arguments

res_list	A named list of enrichment result objects (e.g., enrichResult, gseaResult, nseaResult).
method	Character, aggregation method for p-values. One of "brown", "fisher", or "stouffer".
...	Additional arguments passed to aggregate_omics (e.g., cov_matrix for Brown's method).

Value

An enrichResult object containing the aggregated p-values, FDR, and combined gene lists.

aggregate_omics	<i>Aggregate multi-omics gene/protein-level statistics</i>
-----------------	--

Description

Aggregate multi-omics or multi-source statistics into a unified object for downstream enrichment analysis.

Usage

```
aggregate_omics(
  x,
  method = c("fisher", "stouffer", "brown", "mean", "weighted_mean", "max_abs"),
  input = c("pvalue", "signed_score"),
  feature_type = "gene",
  conflict_policy = c("keep_all", "strict", "penalty"),
  ...
)
```

Arguments

x	A list of named numeric vectors, a data.frame, or a matrix. Row names (or names for vectors) must represent feature IDs.
method	Character, aggregation method. One of "fisher", "stouffer", "mean", or "max_abs".
input	Character, input type. One of "pvalue" or "signed_score".
feature_type	Character, type of the features (e.g., "gene", "protein"). Default is "gene".
conflict_policy	Character, strategy to handle directional conflicts when input is "signed_score". One of "keep_all" (default, ignore conflicts), "strict" (set to NA if any signs conflict), or "penalty" (divide final score by 2 if signs conflict).
...	Additional arguments.

Value

An object of class `omics_aggregated` containing `score`, `pvalue` (if input is "pvalue"), `input_type`, `feature_type`, and `feature_id`.

bayes_enrich

Bayesian term selection for enrichment results

Description

`bayes_enrich()` adds a model-based selection layer on top of ORA results. It estimates the posterior probability that each candidate term is an active biological program explaining the observed input genes.

Usage

```
bayes_enrich(
  x,
  candidate = c("top", "significant", "all"),
  n_terms = 200,
  by = "p.adjust",
  prior = 0.1,
  false_positive = 0.01,
  false_negative = 0.1,
  n_iter = 5000,
  burnin = 1000,
  thin = 1,
  posterior_cutoff = 0.5,
  seed = NULL,
  verbose = FALSE
)
```

Arguments

<code>x</code>	An <code>enrichResult</code> object, typically from <code>ora_gson()</code> or a package that builds on <code>enrichit</code> , such as <code>clusterProfiler</code> .
<code>candidate</code>	Candidate terms to include. "significant" uses <code>as.data.frame(x)</code> , "all" uses <code>x@result</code> , "top" uses the top <code>n_terms</code> rows from <code>x@result</code> ordered by <code>by</code> ; or provide a character vector of term IDs.
<code>n_terms</code>	Maximum number of candidate terms when <code>candidate = "top"</code> or when more candidates are supplied than this value. Use <code>Inf</code> to disable.
<code>by</code>	Column used to order candidate terms.
<code>prior</code>	Prior probability that a term is active.
<code>false_positive</code>	Probability of observing a gene not covered by active terms.
<code>false_negative</code>	Probability of missing a gene covered by active terms.

n_iter	Total number of MCMC iterations.
burnin	Number of initial iterations discarded.
thin	Keep one sample every thin iterations after burn-in.
posterior_cutoff	Terms with posterior greater than or equal to this value are marked active.
seed	Optional random seed.
verbose	Print sampler progress.

Details

The implementation uses a lightweight Metropolis-Hastings sampler over binary latent term states. Given active terms, each gene is modeled as observed with probability $1 - \text{false_negative}$ if covered by at least one active term, and with probability false_positive otherwise. The prior probability that a candidate term is active is prior.

This is intended as a result-compression and interpretation layer, not as a replacement for ORA p-values.

Value

The input `enrichResult` object with additional columns in `@result`: `posterior`, `posterior_odds`, `bayes_rank`, `bayes_active`, `bayes_covered_gene`, and `bayes_covered_count`.

bayes_summary	<i>Summarize Bayesian enrichment results</i>
---------------	--

Description

Return a data frame sorted by posterior probability from a result processed by `bayes_enrich()`. This is a convenience wrapper around sorting `as.data.frame(x)` by decreasing posterior.

Usage

```
bayes_summary(x, active = FALSE, n = Inf)
```

Arguments

x	An <code>enrichResult</code> object processed by <code>bayes_enrich()</code> .
active	Logical. If TRUE, keep only rows with <code>bayes_active = TRUE</code> .
n	Number of rows to return. Use <code>Inf</code> to return all rows.

Value

A data frame ordered by decreasing posterior.

```
classify_omics_pattern
```

Classify pathway-level multi-omics patterns

Description

Compare merged enrichment results with single-omics enrichment results to classify the contribution pattern of each pathway.

Usage

```
classify_omics_pattern(
  merged_res,
  single_res,
  p_cutoff = 0.05,
  by = "p.adjust"
)
```

Arguments

<code>merged_res</code>	An <code>enrichResult</code> or <code>gseaResult</code> object from the merged multi-omics analysis.
<code>single_res</code>	A named list of <code>enrichResult</code> or <code>gseaResult</code> objects from single-omics analyses.
<code>p_cutoff</code>	Numeric, the significance cutoff. Default is 0.05.
<code>by</code>	Character, the column to use for significance threshold. Default is "p.adjust".

Value

The `merged_res` object with an additional column `Omics_Pattern` in its result data.frame.

```
collapse_multilayer_scores
```

Collapse multi-layer diffusion scores

Description

Collapse multi-layer diffusion scores

Usage

```
collapse_multilayer_scores(
  x,
  collapse = c("weighted_mean", "sum", "mean", "max_abs"),
  layer_weights = NULL,
  output_space = c("union", "gene"),
  mapping = NULL,
  target_layer = NULL
)
```

Arguments

x result from `propagate_multilayer()`.

collapse one of "weighted_mean", "sum", "mean", or "max_abs".

layer_weights optional named numeric vector used when `collapse = "weighted_mean"`.

output_space one of "union" or "gene".

mapping optional mapping data.frame with `source_id`, `target_id`, and optional layer columns.

target_layer optional layer name to extract before collapsing.

Value

A `multilayer_collapsed` object with a score vector.

`compareClusterResult`-class

Class "compareClusterResult" This class represents the comparison result of gene clusters by GO categories at specific level or GO enrichment analysis.

Description

Class "compareClusterResult" This class represents the comparison result of gene clusters by GO categories at specific level or GO enrichment analysis.

Slots

`compareClusterResult` cluster comparing result

`geneClusters` a list of genes

`fun` one of `groupGO`, `enrichGO` and `enrichKEGG`

`gene2Symbol` gene ID to Symbol

`keytype` Gene ID type

`readable` logical flag of gene ID in symbol or not.

.call function call
 termsim Similarity between term
 method method of calculating the similarity between nodes
 dr dimension reduction result
 organism organism

Author(s)

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

See Also

[enrichResult](#)

enrichResult-class	<i>Class "enrichResult" This class represents the result of enrichment analysis.</i>
--------------------	--

Description

Class "enrichResult" This class represents the result of enrichment analysis.

Slots

result enrichment analysis
 pvalueCutoff pvalueCutoff
 pAdjustMethod pvalue adjust method
 qvalueCutoff qvalueCutoff
 organism only "human" supported
 ontology biological ontology
 gene Gene IDs
 keytype Gene ID type
 universe background gene
 gene2Symbol mapping gene to Symbol
 geneSets gene sets
 readable logical flag of gene ID in symbol or not.
 termsim Similarity between term
 method method of calculating the similarity between nodes
 dr dimension reduction result

Author(s)

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

`EXTID2NAME`*EXTID2NAME*

Description

mapping gene ID to gene Symbol

Usage

```
EXTID2NAME(OrgDb, geneID, keytype, toType = "SYMBOL")
```

Arguments

<code>OrgDb</code>	OrgDb
<code>geneID</code>	entrez gene ID
<code>keytype</code>	keytype
<code>toType</code>	ID type of the output

Value

gene symbol

Author(s)

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

`extract_mnsea_subnetwork`*Extract pathway subnetwork data from a mnseaResult*

Description

Extract pathway subnetwork data from a mnseaResult

Usage

```
extract_mnsea_subnetwork(  
  res,  
  pathway_id = NULL,  
  include_couplings = TRUE,  
  include_isolated = TRUE  
)
```

Arguments

`res` A `mnseaResult` object.

`pathway_id` Optional pathway ID. If `NULL`, the top pathway is used.

`include_couplings` Logical, whether to include inter-layer coupling edges. Default is `TRUE`.

`include_isolated` Logical, whether to keep nodes without retained edges. Default is `TRUE`.

Value

A list with `pathway`, `layer_contribution`, `nodes`, and `edges`.

geneID	<i>geneID generic</i>
--------	-----------------------

Description

geneID generic

Usage

geneID(x)

Arguments

`x` `enrichResult` object

Value

'geneID' return the 'geneID' column of the enriched result which can be converted to `data.frame` via 'as.data.frame'

Examples

```
## Not run:
data(geneList, package="DOSE")
de <- names(geneList)[1:100]
x <- DOSE::enrichDO(de)
geneID(x)

## End(Not run)
```

geneInCategory	<i>geneInCategory generic</i>
----------------	-------------------------------

Description

geneInCategory generic

Usage

```
geneInCategory(x)
```

Arguments

x	enrichResult
---	--------------

Value

'geneInCategory' return a list of genes, by splitting the input gene vector to enriched functional categories

Examples

```
## Not run:
data(geneList, package="DOSE")
de <- names(geneList)[1:100]
x <- DOSE::enrichDO(de)
geneInCategory(x)

## End(Not run)
```

```
get_mnsea_contribution
```

Get cached contribution tables from a mnseaResult

Description

Get cached contribution tables from a mnseaResult

Usage

```
get_mnsea_contribution(res, pathway_id = NULL, level = c("pathway", "feature"))
```

Arguments

res	A mnseaResult object.
pathway_id	Optional pathway ID. If NULL, returns all pathways for level = "pathway" and uses the top pathway for level = "feature".
level	One of "pathway" or "feature".

Value

A data.frame containing cached contribution information.

get_omics_contribution	<i>Get gene-level omics contribution for a specific pathway</i>
------------------------	---

Description

Extract the original multi-omics statistics for genes in a specific enriched pathway.

Usage

```
get_omics_contribution(res, agg, pathway_id = NULL)
```

Arguments

- res An enrichResult or gseaResult object.
- agg An omics_aggregated object from aggregate_omics().
- pathway_id Character, the ID of the pathway to extract. If NULL, the top pathway is used.

Value

A data.frame containing the genes, their original omics statistics, the aggregated score, and whether they belong to the core enrichment.

gsea	<i>Gene Set Enrichment Analysis (GSEA)</i>
------	--

Description

Perform Gene Set Enrichment Analysis (GSEA) using a ranked gene list.

Usage

```
gsea(  
  geneList,  
  gene_sets,  
  weight = NULL,  
  minGSSize = 10,  
  maxGSSize = 500,  
  nPerm = 1000,  
  exponent = 1,  
  method = "multilevel",  
  adaptive = FALSE,
```

```

minPerm = 101,
maxPerm = 1e+05,
pvalThreshold = 0.1,
eps = 1e-10,
sampleSize = 101,
seed = FALSE,
nPermSimple = 1000,
scoreType = "std",
verbose = TRUE
)

```

Arguments

geneList	A named numeric vector of gene statistics (e.g., log fold change), ranked in descending order.
gene_sets	A named list of gene sets. Each element is a character vector of genes.
weight	A named numeric vector of weights for genes. The names should match the names of geneList. If provided, the geneList will be multiplied by the weight and resorted before GSEA (default: NULL).
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of each geneSet for analyzing
nPerm	Number of permutations for p-value calculation (default: 1000).
exponent	Weighting exponent for enrichment score (default: 1.0).
method	Permutation method.
adaptive	Logical. Use adaptive permutation.
minPerm	Minimum permutations for adaptive mode.
maxPerm	Maximum permutations for adaptive mode.
pvalThreshold	P-value threshold for early stopping.
eps	Epsilon for multilevel methods (default: 1e-10). Sets the smallest p-value that can be estimated.
sampleSize	Sample size for multilevel methods (default: 101).
seed	Random seed for reproducibility (default: FALSE). Set to a number to make the result reproducible; set to TRUE to use a fixed default seed. If FALSE (default), a seed is drawn from R's random number generator on each run, so results may vary between runs. Note that the C++ engine uses its own RNG seeded with this value: passing an explicit numeric seed (or calling <code>set.seed()</code> before <code>gsea()</code> when <code>seed = FALSE</code>) makes the result reproducible.
nPermSimple	Number of permutations for the simple method (default: 1000).
scoreType	Type of enrichment score calculation: "std", "pos", "neg" (default: "std").
verbose	Logical. Print progress messages.

Value

A data.frame with columns:

- **ID**: Gene set name
- **enrichmentScore**: Enrichment Score
- **NES**: Normalized Enrichment Score
- **pvalue**: Empirical p-value from permutation test
- **setSize**: Size of the gene set (number of genes found in geneList)
- **nPerm**: (adaptive mode only) Actual number of permutations used
- **rank**: Rank at which the maximum enrichment score is attained
- **leading_edge**: Leading edge statistics (tags, list, signal)
- **core_enrichment**: Genes in the leading edge, separated by '/'

Examples

```
# Example data
stats <- rnorm(1000)
names(stats) <- paste0("Gene", 1:1000)
stats <- sort(stats, decreasing = TRUE)

gs1 <- paste0("Gene", 1:50)
gs2 <- paste0("Gene", 500:550)
gene_sets <- list(Pathway1 = gs1, Pathway2 = gs2)

# Use default fixed permutation method
result <- gsea(geneList=stats, gene_sets=gene_sets, nPerm=100)

# Use adaptive permutation for more accurate p-values

result_adaptive <- gsea(geneList=stats, gene_sets=gene_sets, adaptive=TRUE)
```

gseaResult-class

Class "gseaResult" This class represents the result of GSEA analysis

Description

Class "gseaResult" This class represents the result of GSEA analysis

Slots

```
result  GSEA analysis
organism organism
setType setType
```

geneSets geneSets
 geneList order rank geneList
 keytype ID type of gene
 permScores permutation scores
 params parameters
 gene2Symbol gene ID to Symbol
 readable whether convert gene ID to symbol
 dr dimension reduction result

Author(s)

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

gseaScores

Calculate GSEA Running Enrichment Scores

Description

Calculate GSEA Running Enrichment Scores

Usage

```
gseaScores(geneList, geneSet, exponent = 1, fortify = FALSE)
```

Arguments

geneList	a named numeric vector of gene statistics (e.g., t-statistics or log-fold changes), sorted in decreasing order.
geneSet	a character vector of gene IDs belonging to the gene set.
exponent	a numeric value defining the weight of the running enrichment score. Default is 1.
fortify	logical. If TRUE, returns a data frame with columns x, runningScore, and position. If FALSE (default), returns the enrichment score (ES).

Value

If `fortify = TRUE`, a data frame containing the running enrichment scores and positions. If `fortify = FALSE`, a numeric value representing the Enrichment Score (ES).

Author(s)

Guangchuang Yu

gsea_gson

*gsea_gson***Description**

generic function for gene set enrichment analysis

Usage

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

Arguments

<code>geneList</code>	A named numeric vector of gene statistics (e.g., log fold change), ranked in descending order.
<code>gson</code>	A GSON object containing gene set information.
<code>weight</code>	A named numeric vector of weights for genes.
<code>nPerm</code>	Number of permutations for p-value calculation (default: 1000).
<code>exponent</code>	Weighting exponent for enrichment score (default: 1.0).
<code>minGSSize</code>	minimal size of each geneSet for analyzing
<code>maxGSSize</code>	maximal size of each geneSet for analyzing
<code>pvalueCutoff</code>	P-value cutoff applied to both the raw p-value and the adjusted p-value (<code>p.adjust</code>), consistent with the historical clusterProfiler/DOSE behavior (default: 0.05).
<code>pAdjustMethod</code>	P-value adjustment method (e.g., "BH").
<code>method</code>	Permutation method.
<code>adaptive</code>	Logical. Use adaptive permutation.

minPerm	Minimum permutations for adaptive mode.
maxPerm	Maximum permutations for adaptive mode.
pvalThreshold	P-value threshold for early stopping.
seed	Random seed for reproducibility. See gsea() for details (default: FALSE).
verbose	Logical. Print progress messages.
...	Additional parameters passed to gsea()

Value

gseaResult object

Author(s)

Guangchuang Yu

gsfilter

gsfilter

Description

filter enriched result by gene set size or gene count

Usage

```
gsfilter(x, by = "GSSize", min = NA, max = NA)
```

Arguments

x	instance of enrichResult or compareClusterResult
by	one of 'GSSize' or 'Count'
min	minimal size
max	maximal size

Value

update object

Author(s)

Guangchuang Yu

harmonize_ids	<i>Harmonize feature IDs to a target space</i>
---------------	--

Description

Map protein-level or other feature-level statistics to a unified gene-level space.

Usage

```
harmonize_ids(
  x,
  mapping,
  from = "protein",
  to = "gene",
  collapse = c("max_abs", "mean", "min_p")
)
```

Arguments

x	A structured result from aggregate_omics().
mapping	A data.frame with source_id and target_id columns.
from	Character, source feature type. Default is "protein".
to	Character, target feature type. Default is "gene".
collapse	Character, method to collapse multiple source IDs mapped to a single target ID. One of "max_abs", "mean", or "min_p".

Value

A harmonized omics_aggregated object.

mnsea	<i>Multi-layer Network-based Gene Set Enrichment Analysis</i>
-------	---

Description

Multi-layer Network-based Gene Set Enrichment Analysis

Usage

```
mnsea(
  seed_list,
  networks,
  couplings,
  gene_sets,
  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>gene_sets</code>	list of gene sets.
<code>mode</code>	one of "evidence" or "signed".
<code>layer_weights</code>	optional named numeric vector.
<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" or "gene".
<code>p</code>	restart probability.
<code>interlayer_strength</code>	global scaling factor for coupling edges.
<code>specific_weight</code>	logical.
<code>minGSSize</code>	minimal size of each gene set.
<code>maxGSSize</code>	maximal size of genes annotated for testing.
<code>threshold</code>	convergence threshold.
<code>maxIter</code>	maximal number of iterations.
<code>verbose</code>	logical.
<code>...</code>	additional arguments passed to <code>gsea()</code> .

Value

A mnseaResult object.

mnseaResult-class	<i>Class "mnseaResult" This class represents the result of multi-layer Network-based Set Enrichment Analysis.</i>
-------------------	---

Description

Class "mnseaResult" This class represents the result of multi-layer Network-based Set Enrichment Analysis.

Slots

result enrichment analysis
 organism organism label for the enrichment result
 setType gene set collection type
 geneSets gene sets
 geneList order rank geneList
 keytype ID type of gene
 permScores permutation score matrix inherited from gseaResult
 gene2Symbol gene ID to symbol mapping
 readable logical flag of gene ID in symbol or not.
 termsim Calculation matrix of termsim.
 method Method of termsim.
 params parameters
 dr dimension reduction result
 multilayer_network prepared multi-layer network object.
 layer_scores list of layer-specific diffusion score vectors.
 collapsed_scores numeric vector used for downstream enrichment.
 layer_weights numeric vector of layer weights.
 coupling_table data.frame of inter-layer couplings.
 mode character, "evidence" or "signed".
 iterations integer, the actual number of iterations RWR took to converge.
 restart_prob numeric, the restart probability used in RWR.
 collapse_method character collapse method used on layer scores.
 target_layer optional layer name used for downstream export.
 output_space character output space of collapsed scores.
 pathway_contribution pathway-by-layer contribution table precomputed for explanation.
 feature_contribution feature-by-layer contribution table precomputed for explanation.

Author(s)

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

mnsea_gson

Multi-layer NSEA using a GSON object

Description

Multi-layer NSEA using a GSON object

Usage

```
mnsea_gson(
  seed_list,
  networks,
  couplings,
  gson,
  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.
gson	a GSON object.
mode	one of "evidence" or "signed".
layer_weights	optional named numeric vector.
collapse	one of "weighted_mean", "sum", "mean", or "max_abs".
target_layer	optional layer name to export scores from.
output_space	one of "union" or "gene".

p restart probability.
interlayer_strength global scaling factor for coupling edges.
specific_weight logical.
minGSSize minimal size of each gene set.
maxGSSize maximal size of genes annotated for testing.
threshold convergence threshold.
maxIter maximal number of iterations.
verbose logical.
... additional arguments passed to `gsea_gson()`.

Value

A `mnseaResult` object.

nsea

Network-based Gene Set Enrichment Analysis

Description

Network-based Gene Set Enrichment Analysis

Usage

```

nsea(
  geneList,
  network,
  gene_sets,
  mode = c("evidence", "signed"),
  p = 0.5,
  specific_weight = FALSE,
  minGSSize = 10,
  maxGSSize = 500,
  threshold = 1e-09,
  maxIter = 100,
  significance = c("whole_pipeline", "internal"),
  nPerm = 1000,
  seed = NULL,
  exponent = 1,
  verbose = TRUE,
  ...
)

```

Arguments

geneList	named numeric vector. In "evidence" mode, must be non-negative. In "signed" mode, can contain both positive and negative values.
network	edge list (data.frame) or sparse matrix.
gene_sets	list of gene sets.
mode	character, either "evidence" (default) or "signed". If "signed", the network propagation runs separately for positive and negative values.
p	restart probability for RWR (default is 0.5).
specific_weight	logical, whether to apply gene specificity weighting (TF-IDF style) based on gene frequencies in gene_sets. Default is FALSE.
minGSSize	minimal size of each gene set for analyzing. default here is 10.
maxGSSize	maximal size of genes annotated for testing. default here is 500.
threshold	convergence threshold for RWR (default is 1e-9).
maxIter	maximal number of RWR iterations (default is 100).
significance	character, one of "whole_pipeline" (default) or "internal". "whole_pipeline" computes p-values and NES from a permutation null that re-runs the full pipeline (permute gene labels -> re-diffuse -> recompute enrichment scores) and is statistically calibrated. "internal" runs GSEA's internal label-permutation test on the diffused scores; this is the legacy behaviour and is anti-conservative (it ignores the correlation induced by network diffusion) — use it only for comparison/debugging.
nPerm	number of whole-pipeline permutations used to estimate p-values and NES (default: 1000). The smallest estimable p-value is $1/(nPerm + 1)$.
seed	random seed for the permutation null (default: NULL, use the current R RNG state). Set a numeric seed for reproducible results.
exponent	weight of each step in the enrichment score (default: 1).
verbose	logical, print messages.
...	for significance = "internal", other arguments passed to gsea() (e.g. nPerm, method, nPermSimple). For significance = "whole_pipeline" additional arguments are ignored.

Value

A nseaResult object of NSEA results.

nseaResult-class	<i>Class "nseaResult" This class represents the result of Network-based Set Enrichment Analysis (NSEA).</i>
------------------	---

Description

Class "nseaResult" This class represents the result of Network-based Set Enrichment Analysis (NSEA).

Slots

result enrichment analysis
 organism organism label for the enrichment result
 setType gene set collection type
 geneSets gene sets
 geneList order rank geneList
 keytype ID type of gene
 permScores permutation score matrix inherited from gseaResult
 gene2Symbol gene ID to symbol mapping
 readable logical flag of gene ID in symbol or not.
 termsim Calculation matrix of termsim.
 method Method of termsim.
 params parameters
 dr dimension reduction result
 network sparse matrix or data.frame representing the underlying network.
 diffusion_scores numeric vector of RWR diffusion scores for each node.
 mode character, "evidence" or "signed", describing the RWR propagation mode.
 iterations integer, the actual number of iterations RWR took to converge.
 restart_prob numeric, the restart probability used in RWR.

Author(s)

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

nsea_gson

*Network-based GSEA using a GSON object***Description**

Network-based GSEA using a GSON object

Usage

```
nsea_gson(
  geneList,
  network,
  gson,
  mode = c("evidence", "signed"),
  p = 0.5,
  specific_weight = FALSE,
  minGSSize = 10,
  maxGSSize = 500,
  threshold = 1e-09,
  maxIter = 100,
  significance = c("whole_pipeline", "internal"),
  nPerm = 1000,
  seed = NULL,
  exponent = 1,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  verbose = TRUE,
  ...
)
```

Arguments

geneList	named numeric vector. In "evidence" mode, must be non-negative. In "signed" mode, can contain both positive and negative values.
network	edge list (data.frame) or sparse matrix.
gson	a GSON object.
mode	character, either "evidence" (default) or "signed".
p	restart probability for RWR (default is 0.5).
specific_weight	logical, whether to apply gene specificity weighting (TF-IDF style) based on gene frequencies in the GSON object. Default is FALSE.
minGSSize	minimal size of each gene set for analyzing. default here is 10.
maxGSSize	maximal size of genes annotated for testing. default here is 500.
threshold	convergence threshold for RWR (default is 1e-9).

maxIter	maximal number of RWR iterations (default is 100).
significance	character, one of "whole_pipeline" (default) or "internal". "whole_pipeline" computes p-values and NES from a permutation null that re-runs the full pipeline (permute gene labels -> re-diffuse -> recompute enrichment scores) and is statistically calibrated. "internal" runs GSEA's internal label-permutation test on the diffused scores; this is the legacy behaviour and is anti-conservative (it ignores the correlation induced by network diffusion) — use it only for comparison/debugging.
nPerm	number of whole-pipeline permutations used to estimate p-values and NES (default: 1000). The smallest estimable p-value is $1/(nPerm + 1)$.
seed	random seed for the permutation null (default: NULL, use the current R RNG state). Set a numeric seed for reproducible results.
exponent	weight of each step in the enrichment score (default: 1).
pvalueCutoff	p-value cutoff applied to both raw and adjusted p-values (default: 0.05). Set to 1 to retain all tested gene sets.
pAdjustMethod	p-value adjustment method (default: "BH").
verbose	logical, print messages.
...	for significance = "internal", other arguments passed to gsea_gson(); ignored for significance = "whole_pipeline".

Value

A nseaResult object.

ora	<i>Over-Representation Analysis (ORA)</i>
-----	---

Description

Perform over-representation analysis using hypergeometric test (Fisher's exact test).

Usage

```
ora(gene, gene_sets, universe, weight = NULL)
```

Arguments

gene	Character vector of differentially expressed genes (or gene list of interest).
gene_sets	A named list of gene sets. Each element is a character vector of genes.
universe	Character vector of background genes (e.g., all genes in the platform).
weight	A named numeric vector of weights for background genes. If provided, Weighted ORA will be performed using Wallenius' noncentral hypergeometric distribution (requires 'BiasedUrn' package). The names should match the universe genes.

Value

A data.frame with columns:

GeneSet	Gene set name
SetSize	Number of genes in the gene set (intersected with universe)
DEInSet	Number of differentially expressed genes in the gene set
DESize	Total number of differentially expressed genes in universe
PValue	Raw p-value from hypergeometric test

Examples

```
# Example data
de_genes <- c("Gene1", "Gene2", "Gene3", "Gene4", "Gene5")
all_genes <- paste0("Gene", 1:1000)

gs1 <- paste0("Gene", 1:50)
gs2 <- paste0("Gene", 51:150)
gs3 <- paste0("Gene", 151:300)
gene_sets <- list(Pathway1 = gs1, Pathway2 = gs2, Pathway3 = gs3)

result <- ora(gene=de_genes, gene_sets=gene_sets, universe=all_genes)
head(result)
```

ora_gson	<i>ora-gson</i>
----------	-----------------

Description

internal method for enrichment analysis

Usage

```
ora_gson(
  gene,
  pvalueCutoff,
  pAdjustMethod = "BH",
  universe = NULL,
  weight = NULL,
  minGSSize = 10,
  maxGSSize = 500,
  qvalueCutoff = 0.2,
  gson
)
```

Arguments

gene	a vector of entrez gene id.
pvalueCutoff	P-value cutoff.
pAdjustMethod	P-value adjustment method (e.g., "BH").
universe	background genes, default is the intersection of the 'universe' with genes that have annotations. Users can set options(enrichment_force_universe = TRUE) to force the 'universe' untouched.
weight	A named numeric vector of weights for background genes. If provided, Weighted ORA will be performed.
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of each geneSet for analyzing
qvalueCutoff	cutoff of qvalue
gson	A GSON object containing gene set information.

Details

using the hypergeometric model

Value

A enrichResult instance.

Author(s)

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

```
prepare_multilayer_network
```

Prepare multi-layer network for repeated propagation

Description

Prepare multi-layer network for repeated propagation

Usage

```
prepare_multilayer_network(
  networks,
  couplings,
  directed = FALSE,
  intra_normalize = "column",
  inter_normalize = "column",
  interlayer_strength = 1,
  layer_order = names(networks)
)
```

Arguments

networks	named list of layer-specific networks.
couplings	data.frame of inter-layer edges with columns from_layer, from_id, to_layer, to_id, and optional weight.
directed	logical, whether the multi-layer graph is directed.
intra_normalize	one of "column", "row", or "none".
inter_normalize	one of "column", "row", or "none".
interlayer_strength	numeric scalar used to scale all coupling edges.
layer_order	explicit layer order. Defaults to names(networks).

Value

A multilayer_network object.

prepare_network	<i>Prepare network for repeated NSEA runs</i>
-----------------	---

Description

Prepare network for repeated NSEA runs

Usage

```
prepare_network(network, directed = FALSE, normalize = "column")
```

Arguments

network	edge list (data.frame with 2 or 3 columns), a sparse matrix, or a mechgraph object. For mechgraph, its edges table must have from/to columns; a numeric score or weight column, if present, is used as the edge weight, otherwise unit weights are used.
directed	logical, whether the network is directed. Default is FALSE.
normalize	one of "column", "row", or "none". Default is "column".

Value

A sparse matrix (dgCMatrix) that has been properly formatted and normalized.

propagate_multilayer *Propagate signals on a multi-layer network*

Description

Propagate signals on a multi-layer network

Usage

```
propagate_multilayer(  
  seed_list,  
  network,  
  mode = c("evidence", "signed"),  
  p = 0.5,  
  threshold = 1e-09,  
  maxIter = 100,  
  layer_weights = NULL,  
  target_layer = NULL  
)
```

Arguments

seed_list	named list of named numeric vectors, one per layer.
network	a prepared multilayer_network object.
mode	one of "evidence" or "signed".
p	restart probability.
threshold	convergence threshold.
maxIter	maximum number of iterations.
layer_weights	optional named numeric vector of layer weights.
target_layer	optional layer name to focus on downstream.

Value

A multilayer_propagation object.

select_features_for_ora	
	<i>Select features for ORA</i>

Description

Convert continuous aggregated statistics into a discrete list of genes and a universe for Over-Representation Analysis.

Usage

```
select_features_for_ora(x, cutoff = 0.05, by = c("pvalue", "score"), ...)
```

Arguments

x	A structured result from aggregate_omics() or harmonize_ids().
cutoff	Numeric, the threshold to apply.
by	Character, metric to apply the threshold on. One of "pvalue" or "score".
...	Additional arguments.

Value

A list containing gene (the selected feature IDs) and universe (all feature IDs).

setReadable	<i>setReadable</i>
-------------	--------------------

Description

mapping geneID to gene Symbol

Usage

```
setReadable(x, OrgDb, keyType = "auto", toType = "SYMBOL")
```

Arguments

x	enrichResult Object
OrgDb	OrgDb
keyType	keyType of gene
toType	ID type of the output

Value

enrichResult Object

Author(s)

Guangchuang Yu

show	<i>show method</i>
------	--------------------

Description

- show method for gseaResult instance
- show method for nseaResult instance
- show method for mnseaResult instance
- show method for enrichResult instance

Usage

- show(object)
- show(object)
- show(object)
- show(object)

Arguments

- object A enrichResult instance.

Value

- message
- message
- message
- message

Author(s)

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

summary	<i>summary method</i>
---------	-----------------------

Description

summary method for gseaResult instance
summary method for enrichResult instance

Usage

summary(object, ...)

summary(object, ...)

Arguments

object	A enrichResult instance.
...	additional parameter

Value

A data frame
A data frame

Author(s)

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

Index

- * **classes**
 - compareClusterResult-class, [7](#)
 - enrichResult-class, [8](#)
 - gseaResult-class, [14](#)
- * **manip**
 - ora_gson, [27](#)
- aggregate_enrichment, [2](#)
- aggregate_omics, [3](#)
- bayes_enrich, [4](#)
- bayes_summary, [5](#)
- classify_omics_pattern, [6](#)
- collapse_multilayer_scores, [6](#)
- compareClusterResult-class, [7](#)
- enrichResult, [8](#)
- enrichResult-class, [8](#)
- EXTID2NAME, [9](#)
- extract_mnsea_subnetwork, [9](#)
- geneID, [10](#)
- geneInCategory, [11](#)
- get_mnsea_contribution, [11](#)
- get_omics_contribution, [12](#)
- gsea, [12](#)
- gsea_gson, [16](#)
- gseahResult-class (gseaResult-class), [14](#)
- gseaResult-class, [14](#)
- gseaScores, [15](#)
- gsfilter, [17](#)
- harmonize_ids, [18](#)
- mnsea, [18](#)
- mnsea_gson, [21](#)
- mnseaResult-class, [20](#)
- nsea, [22](#)
- nsea_gson, [25](#)
- nseaResult-class, [24](#)
- ora, [26](#)
- ora_gson, [27](#)
- plot, compareClusterResult-method
(compareClusterResult-class), [7](#)
- prepare_multilayer_network, [28](#)
- prepare_network, [29](#)
- propagate_multilayer, [30](#)
- select_features_for_ora, [31](#)
- setReadable, [31](#)
- show, [32](#)
- show, compareClusterResult-method
(compareClusterResult-class), [7](#)
- show, enrichResult-method
(enrichResult-class), [8](#)
- show, gseaResult-method
(gseaResult-class), [14](#)
- show, mnseaResult-method (show), [32](#)
- show, nseaResult-method (show), [32](#)
- summary, [33](#)
- summary, compareClusterResult-method
(compareClusterResult-class), [7](#)
- summary, enrichResult-method
(enrichResult-class), [8](#)
- summary, gseaResult-method
(gseaResult-class), [14](#)