

Package ‘enrichit’

June 16, 2026

Title 'C++' Implementations of Functional Enrichment Analysis

Version 0.1.5

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Description Fast implementations of functional enrichment analysis methods using 'C++' via 'Rcpp'.
Currently provides Over-Representation Analysis (ORA) and Gene Set Enrichment Analysis (GSEA).
The multilevel GSEA algorithm is derived from the 'fgsea' package.
Methods are described in Subramanian et al. (2005) <doi:10.1073/pnas.0506580102> and Korotkevich et al. (2021) <doi:10.1101/060012>.

License Artistic-2.0

Depends R (>= 3.5.0)

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

LinkingTo Rcpp

Suggests AnnotationDbi, clusterProfiler, DOSE, gson, qvalue, quarto, testthat

Encoding UTF-8

VignetteBuilder quarto

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Guangchuang Yu [aut, cre]

Repository CRAN

Date/Publication 2026-06-16 07:30:02 UTC

Contents

bayes_enrich	2
bayes_summary	3
compareClusterResult-class	4
enrichit_params	5
enrichResult-class	5
EXTID2NAME	6

geneID	7
geneInCategory	7
gsea	8
gseaResult-class	10
gseaScores	10
gsea_gson	11
gsfilter	12
ora	13
ora_gson	14
setReadable	15
show	15
summary	16
Index	17

bayes_enrich	<i>Bayesian term selection for enrichment results</i>
--------------	---

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

x An enrichResult object, typically from ora_gson() or a package that builds on enrichit, such as clusterProfiler.

candidate	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.
n_terms	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.
by	Column used to order candidate terms.
prior	Prior probability that a term is active.
false_positive	Probability of observing a gene not covered by active terms.
false_negative	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 <code>thin</code> 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`.

<code>bayes_summary</code>	<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 `as.data.frame(x)` by decreasing posterior.

Usage

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

Arguments

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

Value

A data frame ordered by decreasing posterior.

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](#)

enrichit_params	<i>Common parameters for enrichit functions</i>
-----------------	---

Description

Common parameters for enrichit functions

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.
nPerm	Number of permutations for p-value calculation (default: 1000).
exponent	Weighting exponent for enrichment score (default: 1.0).
minGSSize	minimal size of each geneSet for analyzing
maxGSSize	maximal size of each geneSet for analyzing
pvalueCutoff	P-value cutoff.
pAdjustMethod	P-value adjustment method (e.g., "BH").
verbose	Logical. Print progress messages.
gson	A GSON object containing gene set information.
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.

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	<i>EXTID2NAME</i>
------------	-------------------

Description

mapping gene ID to gene Symbol

Usage

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

Arguments

OrgDb	OrgDb
geneID	entrez gene ID
keytype	keytype
toType	ID type of the output

Value

gene symbol

Author(s)

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

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

```
data(geneList, package="DOSE")
de <- names(geneList)[1:100]
x <- DOSE::enrichDO(de)
geneID(x)
```

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

```
data(geneList, package="DOSE")
de <- names(geneList)[1:100]
x <- DOSE::enrichDO(de)
geneInCategory(x)
```

gsea

Gene Set Enrichment Analysis (GSEA)

Description

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

Usage

```
gsea(
  geneList,
  gene_sets,
  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.
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). If FALSE, a random seed is generated.
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	<i>Class "gseaResult" This class represents the result of GSEA analysis</i>
------------------	---

Description

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

Slots

result GSEA anaysis
 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	<i>Calculate GSEA Running Enrichment Scores</i>
------------	---

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

<i>gsea_gson</i>	<i>gsea_gson</i>
------------------	------------------

Description

generic function for gene set enrichment analysis

Usage

```
gsea_gson(  
  geneList,  
  gson,  
  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,  
  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>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.

pAdjustMethod	P-value adjustment method (e.g., "BH").
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.
verbose	Logical. Print progress messages.
...	Additional parameters passed to gsea()

Value

gseaResult object

Author(s)

Guangchuang Yu

<i>gsfilter</i>	<i>gsfilter</i>
-----------------	-----------------

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

`ora`*Over-Representation Analysis (ORA)*

Description

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

Usage

```
ora(gene, gene_sets, universe)
```

Arguments

<code>gene</code>	Character vector of differentially expressed genes (or gene list of interest).
<code>gene_sets</code>	A named list of gene sets. Each element is a character vector of genes.
<code>universe</code>	Character vector of background genes (e.g., all genes in the platform).

Value

A data.frame with columns:

<code>GeneSet</code>	Gene set name
<code>SetSize</code>	Number of genes in the gene set (intersected with universe)
<code>DEInSet</code>	Number of differentially expressed genes in the gene set
<code>DESize</code>	Total number of differentially expressed genes in universe
<code>PValue</code>	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

ora-gson

Description

interal method for enrichment analysis

Usage

```
ora_gson(
  gene,
  pvalueCutoff,
  pAdjustMethod = "BH",
  universe = 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.
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>

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 enrichResult instance

Usage

- show(object)
- show(object)

Arguments

- | | |
|--------|--------------------------|
| object | A enrichResult instance. |
|--------|--------------------------|

Value

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, [4](#)
 - enrichResult-class, [5](#)
 - gseaResult-class, [10](#)
- * **manip**
 - ora_gson, [14](#)
- bayes_enrich, [2](#)
- bayes_summary, [3](#)
- compareClusterResult-class, [4](#)
- enrichit_params, [5](#)
- enrichResult, [4](#)
- enrichResult-class, [5](#)
- EXTID2NAME, [6](#)
- geneID, [7](#)
- geneInCategory, [7](#)
- gsea, [8](#)
- gsea_gson, [11](#)
- gseahResult-class (gseaResult-class), [10](#)
- gseaResult-class, [10](#)
- gseaScores, [10](#)
- gsfilter, [12](#)
- ora, [13](#)
- ora_gson, [14](#)
- plot, compareClusterResult-method
(compareClusterResult-class), [4](#)
- setReadable, [15](#)
- show, [15](#)
- show, compareClusterResult-method
(compareClusterResult-class), [4](#)
- show, enrichResult-method
(enrichResult-class), [5](#)
- show, gseaResult-method
(gseaResult-class), [10](#)
- summary, [16](#)
- summary, compareClusterResult-method
(compareClusterResult-class), [4](#)
- summary, enrichResult-method
(enrichResult-class), [5](#)
- summary, gseaResult-method
(gseaResult-class), [10](#)