Package 'MicrobiomeProfiler'

June 5, 2026

Title An R/shiny package for microbiome functional enrichment analysis

Version 1.19.1

Description
This is an R/shiny package to perform functional enrichment analysis for microbiome data. This package was based on clusterProfiler. Moreover, MicrobiomeProfiler support KEGG enrichment analysis,
COG enrichment analysis, Microbe-Disease association enrichment analysis, Metabo-Pathway analysis.

License GPL-2

Depends R (>= 4.2.0)

Imports clusterProfiler (>= 4.5.2), enrichit, config, DT, enrichplot, golem, gson, methods, magrittr, shiny (>= 1.6.0), shinyWidgets, shinycustomloader, htmltools, ggplot2, graphics, stats, utils, yulab.utils

Suggests rmarkdown, pkgload, knitr, testthat (>= 3.0.0), prettydoc

Encoding UTF-8

biocViews Microbiome, Software, Visualization,KEGG

RoxygenNote 7.3.3

URL https://github.com/YuLab-SMU/MicrobiomeProfiler/, https://yulab-smu.top/contribution-knowledge-mining/

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

Config/testthat/edition 3

VignetteBuilder knitr

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

git_branch devel

git_last_commit 82c128c

git_last_commit_date 2026-05-20

Repository Bioconductor 3.24

Date/Publication 2026-06-04

Author Guangchuang Yu [cre, aut] (ORCID: <https://orcid.org/0000-0002-6485-8781>), Meijun Chen [aut] (ORCID: <https://orcid.org/0000-0003-2486-8106>)

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Contents

`MicrobiomeProfiler-package`

MicrobiomeProfiler: An R/shiny package for microbiome functional enrichment analysis

---

Description

This is an R/shiny package to perform functional enrichment analysis for microbiome data. This package was based on clusterProfiler. Moreover, MicrobiomeProfiler support KEGG enrichment analysis, COG enrichment analysis, Microbe-Disease association enrichment analysis, Metabo-Pathway analysis.

Author(s)

Maintainer: Guangchuang Yu <guangchuangyu@gmail.com> (ORCID)

Authors:

- Meijun Chen <mjchen1996@outlook.com> (ORCID)

See Also

Useful links:

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

---

`enrichCOG` *COG enrichment analysis for microbiome data*

Description

COG enrichment analysis for microbiome data

Usage

```
enrichCOG(
  gene,
  dtype = "category",
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  universe = NULL,
  minGSSize = 10,
  maxGSSize = 500,
  qvalueCutoff = 0.2
)
```

Arguments

`gene`	a vector of COG ids.
`dtype`	one of "category", "pathway"
`pvalueCutoff`	adjusted pvalue cutoff on enrichment tests to report.
`pAdjustMethod`	one of "holm","hochberg","hommel","bonferroni","BH", "BY","fdr","none".
`universe`	universe background genes. If missing,use the all COGs.
`minGSSize`	minimal size of genes annotated by KEGG term for testing.
`maxGSSize`	maximal size of genes annotated for testing.
`qvalueCutoff`	qvalue cutoff on enrichment tests to report.

Value

A `enrichResult` instance.

Examples

```
data(Psoriasis_data)
cog <- enrichCOG(Psoriasis_data,dtype="category")
```

enrichHMDB *Metabolism enrichment analysis for microbiome data*

Description

Metabolism enrichment analysis for microbiome data

Usage

```
enrichHMDB(
  metabo_list,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  universe = NULL,
  minGSSize = 10,
  maxGSSize = 500,
  qvalueCutoff = 0.2
)
```

Arguments

metabo_list	a vector of metabolites in HMDB.ID
pvalueCutoff	adjusted pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
universe	universe background genes. If missing, use HMDB db.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
qvalueCutoff	qvalue cutoff on enrichment tests to report.

Value

A enrichResult instance.

Examples

```
x1 <- c("HMDB0000001","HMDB0000005","HMDB0000008")
x2 <- enrichHMDB(x1)
```

enrichKO *KO enrichment for microbiome data*

Description

KO enrichment for microbiome data

Usage

```
enrichKO(
  gene,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  universe = NULL,
  minGSSize = 10,
  maxGSSize = 500,
  qvalueCutoff = 0.2
)
```

Arguments

gene	a vector of K gene id (e.g. K00001) or EC id (e.g. 1.1.1.27).
pvalueCutoff	adjusted pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm","hochberg","hommel","bonferroni","BH", "BY","fdr","none".
universe	universe background genes. If missing, use all K genes.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
qvalueCutoff	qvalue cutoff on enrichment tests to report.

Value

A enrichResult instance.

Examples

```
data(Rat_data)
ko <- enrichKO(Rat_data)
head(ko)
```

enrichMBKEGG *Metabolism enrichment analysis for microbiome data*

Description

Metabolism enrichment analysis for microbiome data

Usage

```
enrichMBKEGG(
  metabo_list,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  universe = NULL,
  minGSSize = 10,
  maxGSSize = 500,
  qvalueCutoff = 0.2
)
```

Arguments

metabo_list	a vector of metabolites in KEGG.ID
pvalueCutoff	adjusted pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
universe	universe background genes. If missing, use KEGG as default.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
qvalueCutoff	qvalue cutoff on enrichment tests to report.

Value

A enrichResult instance.

Examples

```
mblist3 <- c("C00019","C00020","C00022")
mb3 <- enrichMBKEGG(mblist3)
head(mb3)
```

enrichMDA *Microbe-Disease associations enrichment analysis*

Description

Microbe-Disease associations enrichment analysis

Usage

```
enrichMDA(
  microbe_list,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  universe = NULL,
  minGSSize = 10,
  maxGSSize = 500,
  qvalueCutoff = 0.2
)
```

Arguments

microbe_list	a vector of microbe ncbi tax ids.
pvalueCutoff	adjusted pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
universe	universe background genes. If missing, use disbiome as default.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
qvalueCutoff	qvalue cutoff on enrichment tests to report.

Value

A enrichResult instance.

Examples

```
data(microbiota_taxlist)
mda <- enrichMDA(microbiota_taxlist)
head(mda)
```

enrichModule *Module enrichment for microbiome data*

Description

Module enrichment for microbiome data

Usage

```
enrichModule(
  gene,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  universe = NULL,
  minGSSize = 10,
  maxGSSize = 500,
  qvalueCutoff = 0.2
)
```

Arguments

gene	a vector of K gene id (e.g. K00001).
pvalueCutoff	adjusted pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm","hochberg","hommel","bonferroni","BH", "BY","fdr","none".
universe	universe background genes. If missing, use all K genes.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
qvalueCutoff	qvalue cutoff on enrichment tests to report.

Value

A enrichResult instance.

Examples

```
data(Rat_data)
ko <- enrichModule(Rat_data)
head(ko)
```

enrichSMPDB *Metabolism enrichment analysis for microbiome data*

Description

Metabolism enrichment analysis for microbiome data

Usage

```
enrichSMPDB(
  metabo_list,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  universe = NULL,
  minGSSize = 10,
  maxGSSize = 500,
  qvalueCutoff = 0.2
)
```

Arguments

metabo_list	a vector of metabolites in smpdb Metabolite.ID
pvalueCutoff	adjusted pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
universe	universe background genes. If missing, use SMPDB db.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
qvalueCutoff	qvalue cutoff on enrichment tests to report.

Value

A enrichResult instance.

Examples

```
smp <- enrichSMPDB(c("PW_C000164","PW_C000078","PW_C000040"))
head(smp)
```

gseCOG *COG GSEA enrichment analysis for microbiome data*

Description

COG GSEA enrichment analysis for microbiome data

Usage

```
gseCOG(
  geneList,
  dtype = "category",
  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

geneList	a vector of COG ids with values.
dtype	one of "category", "pathway"
nPerm	number of permutations.
exponent	exponent of weighting.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
pvalueCutoff	pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm","hochberg","hommel","bonferroni","BH", "BY","fdr","none".
method	one of "multilevel", "fgsea", "bioc".
adaptive	whether to use adaptive permutation.
minPerm	minimal number of permutations.
maxPerm	maximal number of permutations.
pvalThreshold	pvalue threshold for adaptive permutation.
verbose	whether to show progress.

Value

A gseaResult instance.

gseHMDB *Metabolism GSEA enrichment analysis for microbiome data*

Description

Metabolism GSEA enrichment analysis for microbiome data

Usage

```
gseHMDB(
  metabo_list,
  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

metabo_list	a vector of metabolites in HMDB.ID with values
nPerm	number of permutations.
exponent	exponent of weighting.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
pvalueCutoff	pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
method	one of "multilevel", "fgsea", "bioc".
adaptive	whether to use adaptive permutation.
minPerm	minimal number of permutations.
maxPerm	maximal number of permutations.
pvalThreshold	pvalue threshold for adaptive permutation.
verbose	whether to show progress.

Value

A gseaResult instance.

gseKO *KO GSEA enrichment for microbiome data*

Description

KO GSEA enrichment for microbiome data

Usage

```
gseKO(
  geneList,
  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

geneList	a vector of K gene id (e.g. K00001) or EC id (e.g. 1.1.1.27) with values.
nPerm	number of permutations.
exponent	exponent of weighting.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
pvalueCutoff	pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm","hochberg","hommel","bonferroni","BH", "BY","fdr","none".
method	one of "multilevel", "fgsea", "bioc".
adaptive	whether to use adaptive permutation.
minPerm	minimal number of permutations.
maxPerm	maximal number of permutations.
pvalThreshold	pvalue threshold for adaptive permutation.
verbose	whether to show progress.

Value

A gseaResult instance.

gseMBKEGG *Metabolism GSEA enrichment analysis for microbiome data*

Description

Metabolism GSEA enrichment analysis for microbiome data

Usage

```
gseMBKEGG(
  metabo_list,
  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

metabo_list	a vector of metabolites in KEGG.ID with values
nPerm	number of permutations.
exponent	exponent of weighting.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
pvalueCutoff	pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm","hochberg","hommel","bonferroni","BH", "BY","fdr","none".
method	one of "multilevel", "fgsea", "bioc".
adaptive	whether to use adaptive permutation.
minPerm	minimal number of permutations.
maxPerm	maximal number of permutations.
pvalThreshold	pvalue threshold for adaptive permutation.
verbose	whether to show progress.

Value

A gseaResult instance.

gseMDA *Microbe-Disease associations GSEA enrichment analysis*

Description

Microbe-Disease associations GSEA enrichment analysis

Usage

```
gseMDA(
  microbe_list,
  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

microbe_list	a vector of microbe ncbi tax ids with values.
nPerm	number of permutations.
exponent	exponent of weighting.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
pvalueCutoff	pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
method	one of "multilevel", "fgsea", "bioc".
adaptive	whether to use adaptive permutation.
minPerm	minimal number of permutations.
maxPerm	maximal number of permutations.
pvalThreshold	pvalue threshold for adaptive permutation.
verbose	whether to show progress.

Value

A gseaResult instance.

gseModule *Module GSEA enrichment for microbiome data*

Description

Module GSEA enrichment for microbiome data

Usage

```
gseModule(
  geneList,
  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

geneList	a vector of K gene id (e.g. K00001) with values.
nPerm	number of permutations.
exponent	exponent of weighting.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
pvalueCutoff	pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm","hochberg","hommel","bonferroni","BH", "BY","fdr","none".
method	one of "multilevel", "fgsea", "bioc".
adaptive	whether to use adaptive permutation.
minPerm	minimal number of permutations.
maxPerm	maximal number of permutations.
pvalThreshold	pvalue threshold for adaptive permutation.
verbose	whether to show progress.

Value

A gseaResult instance.

gseSMPDB *Metabolism GSEA enrichment analysis for microbiome data*

Description

Metabolism GSEA enrichment analysis for microbiome data

Usage

```
gseSMPDB(
  metabo_list,
  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

metabo_list	a vector of metabolites in smpdb Metabolite.ID with values
nPerm	number of permutations.
exponent	exponent of weighting.
minGSSize	minimal size of genes annotated by KEGG term for testing.
maxGSSize	maximal size of genes annotated for testing.
pvalueCutoff	pvalue cutoff on enrichment tests to report.
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
method	one of "multilevel", "fgsea", "bioc".
adaptive	whether to use adaptive permutation.
minPerm	minimal number of permutations.
maxPerm	maximal number of permutations.
pvalThreshold	pvalue threshold for adaptive permutation.
verbose	whether to show progress.

Value

A gseaResult instance.

gson_cpd *gson_cpd*

Description

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

Usage

```
gson_cpd()
```

Value

a 'GSON' object

gson_enzyme *gson_enzyme*

Description

download compound annotation of the latest version of KEGG pathway to enzyme and stored in a 'GSON' object

Usage

```
gson_enzyme()
```

Value

a 'GSON' object

gson_KO *gson_KO*

Description

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

Usage

```
gson_KO()
```

Value

a 'GSON' object

gson_module *gson_module*

Description

download compound annotation of the latest version of KEGG Module and stored in a 'GSON' object

Usage

```
gson_module(db = "ko")
```

Arguments

db	ko or enzyme

Value

a 'GSON' object

microbiota_taxlist *Example data: a vector of 54 bacterial genera tested for significantly between T2D metformin samples*

Description

This example data was reported on Forslund K,et al.2016 (doi: 10.1038/nature15766) and used for Microbe-Disease Association analysis.

Format

a vector with 54 genera tax ids

References

https://www.nature.com/articles/nature15766

Examples

```
data(microbiota_taxlist)
```

Psoriasis_data — *Example data: a vector of 134 significantly different functional COGs between Psoriasis patients and controls*

Description

This example data was reported on Xiao S,et al.2021 (doi:10.3389/fcimb.2021.605825) and used for COG enrichment analysis.

Format

a vector with 134 COGs

Value

https://www.frontiersin.org/articles/10.3389/fcimb.2021.605825/full

Examples

```
data(Psoriasis_data)
```

Rat_data — *Example data: a vector of 91 KEGG Orthologies (KOs) showing significant associations with weaning weight*

Description

This example data was reported on Fang S,et al.2019 (doi: 10.1111/1751-7915.13485) and used for KEGG enrichment analysis.

Format

a vector with 91 KEGG Orthologies (KOs)

References

https://sfamjournals.onlinelibrary.wiley.com/doi/10.1111/1751-7915.13485

Examples

```
data(Rat_data)
```

`run_MicrobiomeProfiler` *Run the Shiny Application*

Description

Run the Shiny Application

Usage

```
run_MicrobiomeProfiler(
  onStart = NULL,
  options = list(),
  enableBookmarking = NULL,
  uiPattern = "/",
  ...
)
```

Arguments

`onStart`	A function that will be called before the app is actually run. This is only needed for `shinyAppObj`, since in the `shinyAppDir` case, a `global.R` file can be used for this purpose.
`options`	Named options that should be passed to the `runApp` call (these can be any of the following: "port", "launch.browser", "host", "quiet", "display.mode" and "test.mode"). You can also specify `width` and `height` parameters which provide a hint to the embedding environment about the ideal height/width for the app.
`enableBookmarking`	Can be one of `"url"`, `"server"`, or `"disable"`. The default value, `NULL`, will respect the setting from any previous calls to `enableBookmarking()`. See `enableBookmarking()` for more information on bookmarking your app.
`uiPattern`	A regular expression that will be applied to each `GET` request to determine whether the `ui` should be used to handle the request. Note that the entire request path must match the regular expression in order for the match to be considered successful.
`...`	arguments to pass to golem_opts. See '?golem::get_golem_options' for more details.

Value

Shiny application object.

Examples

```
if (interactive()) {run_MicrobiomeProfiler()}
```

Index