Package: hypeR (via r-universe)

November 25, 2026

Title An R Package For Geneset Enrichment Workflows

Version 2.12.0

Description An R Package for Geneset Enrichment Workflows.

Depends R (>= 3.6.0)

License GPL-3 + file LICENSE

URL https://github.com/montilab/hypeR

Encoding UTF-8

RoxygenNote 7.3.3

LazyData false

Imports ggplot2, ggforce, R6, magrittr, dplyr, tibble, purrr, stats, methods, utils, stringr, scales, rlang, httr, openxlsx, htmltools, reshape2, reactable, msigdbr, kableExtra, rmarkdown, igraph, visNetwork, shiny, BiocStyle

Suggests tidyverse, devtools, testthat, knitr

biocViews GeneSetEnrichment, Annotation, Pathways

VignetteBuilder knitr

BugReports https://github.com/montilab/hypeR/issues

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

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:54:58 UTC

RemoteUrl https://github.com/bioc/hypeR

RemoteRef RELEASE_3_24

RemoteSha 7c9a3fa9a80f6888fe43be06df1885cea42ebe89

Contents

clean_genesets . . . 3
enrichr_available . . . 3
enrichr_download . . . 4
enrichr_gsets . . . 4
genesets_Server . . . 5
genesets_UI . . . 6
ggempty . . . 6
ggeplot . . . 7
ggvenn . . . 7
gsets . . . 8
hyp . . . 10
hyp_dots . . . 11
hyp_emap . . . 12
hyp_hmap . . . 13
hyp_show . . . 14
hyp_to_excel . . . 15
hyp_to_graph . . . 16
hyp_to_rmd . . . 16
hyp_to_table . . . 18
hypeR . . . 19
hyperdb_available . . . 20
hyperdb_gsets . . . 20
hyperdb_rgsets . . . 21
limma . . . 21
msigdb_available . . . 22
msigdb_download . . . 22
msigdb_gsets . . . 23
msigdb_info . . . 24
msigdb_species . . . 24
msigdb_version . . . 25
multihyp . . . 25
rctbl_build . . . 26
rctbl_hyp . . . 27
rctbl_mhyp . . . 28
rgsets . . . 29
wgcna . . . 31

Index **32**

clean_genesets *Clean labels of genesets*

Description

Clean labels of genesets

Usage

```
clean_genesets(x)
```

Arguments

x	A vector of labels

Examples

```
HALLMARK <- msigdb_download("Homo sapiens", "H")
names(HALLMARK) <- clean_genesets(names(HALLMARK))
head(names(HALLMARK))
```

enrichr_available *Get enrichr available genesets*

Description

Get enrichr available genesets

Usage

```
enrichr_available(
  db = c("Enrichr", "YeastEnrichr", "FlyEnrichr", "WormEnrichr", "FishEnrichr")
)
```

Arguments

db	A species

Value

A dataframe of available genesets

Examples

```
enrichr_available()
```

enrichr_download *Download data from enrichr in the form of a named list*

Description

Download data from enrichr in the form of a named list

Usage

```
enrichr_download(
  genesets,
  db = c("Enrichr", "YeastEnrichr", "FlyEnrichr", "WormEnrichr", "FishEnrichr")
)
```

Arguments

genesets	A name corresponding to available genesets
db	A species

Value

A list of genesets

Examples

```
ATLAS <- enrichr_download("Human_Gene_Atlas")
```

enrichr_gsets *Download data from enrichr in the form of a gsets object*

Description

Download data from enrichr in the form of a gsets object

Usage

```
enrichr_gsets(
  genesets,
  db = c("Enrichr", "YeastEnrichr", "FlyEnrichr", "WormEnrichr", "FishEnrichr"),
  clean = FALSE
)
```

Arguments

genesets	A name corresponding to available genesets
db	A species
clean	Use true to clean labels of genesets

Value

A gsets object

Examples

```
ATLAS <- enrichr_gsets("Human_Gene_Atlas")
```

---

genesets_Server *Shiny server module for geneset selection*

Description

Shiny server module for geneset selection

Usage

```
genesets_Server(id, clean = FALSE)
```

Arguments

id	A unique namespace identifier matching to interface
clean	Use true to clean geneset names

Value

Shiny server code

genesets_UI *Shiny interface module for geneset selection*

Description

Shiny interface module for geneset selection

Usage

```
genesets_UI(id)
```

Arguments

id	A unique namespace identifier

Value

Shiny ui elements

ggempty *An empty ggplot*

Description

An empty ggplot

Usage

```
ggempty()
```

Value

A ggplot object

ggeplot *Enrichment plot implemented in ggplot*

Description

Enrichment plot implemented in ggplot

Usage

```
ggeplot(n, positions, x_axis, y_axis, title = "")
```

Arguments

n	The length of a ranked list
positions	A vector of positions in the ranked list
x_axis	The x-axis of a running enrichment score
y_axis	The y-axis of a running enrichment score
title	Plot title

Value

A ggplot object

ggvenn *Venn diagram implemented in ggplot*

Description

Venn diagram implemented in ggplot

Usage

```
ggvenn(a, b, ga, gb, title = "")
```

Arguments

a	A vector for group a
b	A vector for group b
ga	A string label for group a
gb	A string label for group b
title	Plot title

Value

A ggplot object

gsets *A genesets object*

Description

A genesets object

A genesets object

See Also

rgsets

Public fields

genesets A named list of genesets

name A character vector describing source of genesets

version A character vector describing versioning

Methods

Public methods:

- gsets$new()
- gsets$print()
- gsets$list()
- gsets$info()
- gsets$reduce()
- gsets$clone()

Method new(): Create a gsets object

Usage:

```
gsets$new(
  genesets,
  name = "Custom",
  version = "",
  clean = FALSE,
  quiet = FALSE
)
```

Arguments:

genesets A named list of genesets

name A character vector describing source of genesets

version A character vector describing versioning

clean Use true to clean labels of genesets

quiet Use true to silence warnings

Returns: A new gsets object

Method `print()`**:** Print genesets information

Usage:

```
gsets$print()
```

Returns: NULL

Method `list()`**:** Return genesets as a list

Usage:

```
gsets$list()
```

Returns: A list of genesets

Method `info()`**:** Returns versioning information

Usage:

```
gsets$info()
```

Returns: A character vector with name and version

Method `reduce()`**:** Reduces genesets to a background distribution of symbols

Usage:

```
gsets$reduce(background)
```

Arguments:

`background` A character vector of symbols

Returns: A gsets object

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
gsets$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
genesets <- list("GSET1" = c("GENE1", "GENE2", "GENE3"),
                 "GSET2" = c("GENE4", "GENE5", "GENE6"),
                 "GSET3" = c("GENE7", "GENE8", "GENE9"))

gsets_obj <- gsets$new(genesets, name="example", version="v1.0")
print(gsets_obj)
```

hyp *A hyp object*

Description

A hyp object

A hyp object

See Also

`multihyp`

Public fields

`data` A dataframe returned by hypeR()

`plots` A list of plots returned by hypeR()

`args` A list of arguments passed to hypeR()

`info` Exported information for reproducibility

Methods

Public methods:

- `hyp$new()`
- `hyp$print()`
- `hyp$as.data.frame()`
- `hyp$clone()`

Method `new()`**:** Create a hyp object

Usage:

```
hyp$new(data, plots = NULL, args = NULL, info = NULL)
```

Arguments:

`data` A dataframe returned by hypeR()

`plots` A list of plots returned by hypeR()

`args` A list of arguments passed to hypeR()

`info` Exported information for reproducibility

Returns: A new hyp object

Method `print()`**:** Print hyp obect

Usage:

```
hyp$print()
```

Returns: NULL

Method `as.data.frame()`**:** Extract dataframe from hyp obect

Usage:

```
hyp$as.data.frame()
```

Returns: NULL

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
hyp$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
data <- data.frame(replicate(5,sample(0:1,10,rep=TRUE)))
args <- list("arg_1"=1, "arg_2"=2, "arg_3"=3)
hyp_obj <- hyp$new(data, args=args)
```

`hyp_dots` *Visualize hyp/multihyp objects as a dots plot*

Description

Visualize hyp/multihyp objects as a dots plot

Usage

```
hyp_dots(
  hyp_obj,
  top = 20,
  abrv = 50,
  size_by = c("genesets", "significance", "none"),
  pval = 1,
  fdr = 1,
  val = c("fdr", "pval"),
  title = "",
  merge = FALSE
)
```

Arguments

`hyp_obj`	A hyp or multihyp object
`top`	Limit number of genesets shown
`abrv`	Abbreviation length of geneset labels
`size_by`	Size dots by e.g. c("genesets", "significance", "none")
`pval`	Filter results to be less than pval cutoff

fdr	Filter results to be less than fdr cutoff
val	Choose significance value for plot e.g. c("fdr", "pval")
title	Plot title
merge	Use true to merge a multihyp object into one plot

Value

A ggplot object

Examples

```
genesets <- msigdb_gsets("Homo sapiens", "C2", "CP:KEGG_LEGACY")

signature <- c("IDH3B","DLST","PCK2","CS","PDHB","PCK1","PDHA1","LOC642502",
               "PDHA2","LOC283398","FH","SDHD","OGDH","SDHB","IDH3A","SDHC",
               "IDH2","IDH1","OGDHL","PC","SDHA","SUCLG1","SUCLA2","SUCLG2")

hyp_obj <- hypeR(signature, genesets, background=2522)

hyp_dots(hyp_obj, val="fdr")
```

---

hyp_emap *Visualize hyp/multihyp objects as an enrichment map*

Description

Visualize hyp/multihyp objects as an enrichment map

Usage

```
hyp_emap(
  hyp_obj,
  similarity_metric = c("jaccard_similarity", "overlap_similarity"),
  similarity_cutoff = 0.2,
  pval = 1,
  fdr = 1,
  val = c("fdr", "pval"),
  top = NULL,
  title = ""
)
```

Arguments

hyp_obj	A hyp or multihyp object
similarity_metric	Metric to calculate geneset similarity

similarity_cutoff	Geneset similarity cutoff
pval	Filter results to be less than pval cutoff
fdr	Filter results to be less than fdr cutoff
val	Choose significance value shown above nodes e.g. c("fdr", "pval")
top	Limit number of pathways shown
title	Plot title

Value

A visNetwork object

Examples

```
genesets <- msigdb_gsets("Homo sapiens", "C2", "CP:KEGG_LEGACY")

signature <- c("IDH3B","DLST","PCK2","CS","PDHB","PCK1","PDHA1","LOC642502",
               "PDHA2","LOC283398","FH","SDHD","OGDH","SDHB","IDH3A","SDHC",
               "IDH2","IDH1","OGDHL","PC","SDHA","SUCLG1","SUCLA2","SUCLG2")

hyp_obj <- hypeR(signature, genesets, background=2522)

hyp_emap(hyp_obj, top=30, val="fdr")
```

---

hyp_hmap *Visualize hyp/multihyp objects as a hiearchy map*

Description

Visualize hyp/multihyp objects as a hiearchy map

Usage

```
hyp_hmap(
  hyp_obj,
  pval = 1,
  fdr = 1,
  val = c("fdr", "pval"),
  top = NULL,
  title = "",
  graph = FALSE
)
```

Arguments

hyp_obj	A hyp or multihyp object
pval	Filter results to be less than pval cutoff
fdr	Filter results to be less than fdr cutoff
val	Choose significance value displayed when hovering nodes e.g. c("fdr", "pval")
top	Limit number of pathways shown
title	Plot title
graph	Return an igraph object instead

Value

A visNetwork object

Examples

```
genesets <- hyperdb_rgsets("REACTOME", "70.0")

signature <- c("IDH3B","DLST","PCK2","CS","PDHB","PCK1","PDHA1","LOC642502",
               "PDHA2","LOC283398","FH","SDHD","OGDH","SDHB","IDH3A","SDHC",
               "IDH2","IDH1","OGDHL","PC","SDHA","SUCLG1","SUCLA2","SUCLG2")

hyp_obj <- hypeR(signature, genesets, background=2522)

hyp_hmap(hyp_obj, top=60)
```

---

hyp_show *Convert a hyp object to a reactable table*

Description

Convert a hyp object to a reactable table

Usage

```
hyp_show(hyp_obj, simple = FALSE)
```

Arguments

hyp_obj	A hyp object
simple	Use true to only include essential columns

Value

A reactable table

Examples

```
genesets <- msigdb_gsets("Homo sapiens", "C2", "CP:KEGG_LEGACY")

signature <- c("IDH3B","DLST","PCK2","CS","PDHB","PCK1","PDHA1","LOC642502",
               "PDHA2","LOC283398","FH","SDHD","OGDH","SDHB","IDH3A","SDHC",
               "IDH2","IDH1","OGDHL","PC","SDHA","SUCLG1","SUCLA2","SUCLG2")

hyp_obj <- hypeR(signature, genesets, background=2522)

hyp_show(hyp_obj)
```

---

hyp_to_excel *Export hyp/multihyp object to excel*

Description

Export hyp/multihyp object to excel

Usage

```
hyp_to_excel(
  hyp_obj,
  file_path,
  cols = NULL,
  pval = 1,
  fdr = 1,
  versioning = TRUE
)
```

Arguments

hyp_obj	A hyp or multihyp object
file_path	A file path
cols	Dataframe columns to include
pval	Filter results to p-values less than or equal to this cutoff
fdr	Filter results to FDR values less than or equal to this cutoff
versioning	Add sheet with versioning information

Examples

```
genesets <- msigdb_gsets("Homo sapiens", "C2", "CP:KEGG_LEGACY")

signature <- c("IDH3B","DLST","PCK2","CS","PDHB","PCK1","PDHA1","LOC642502",
               "PDHA2","LOC283398","FH","SDHD","OGDH","SDHB","IDH3A","SDHC",
               "IDH2","IDH1","OGDHL","PC","SDHA","SUCLG1","SUCLA2","SUCLG2")
```

```
hyp_obj <- hypeR(signature, genesets, background=2522)

hyp_to_excel(hyp_obj, file_path="pathways.xlsx")
```

---

hyp_to_graph *Convert a hyp object to an igraph object*

Description

Convert a hyp object to an igraph object

Usage

```
hyp_to_graph(hyp_obj)
```

Arguments

hyp_obj	A hyp object

Value

An igraph object

Examples

```
genesets <- hyperdb_rgsets("REACTOME", "70.0")

signature <- c("IDH3B","DLST","PCK2","CS","PDHB","PCK1","PDHA1","LOC642502",
               "PDHA2","LOC283398","FH","SDHD","OGDH","SDHB","IDH3A","SDHC",
               "IDH2","IDH1","OGDHL","PC","SDHA","SUCLG1","SUCLA2","SUCLG2")

hyp_obj <- hypeR(signature, genesets, background=2522)

ig <- hyp_to_graph(hyp_obj)
```

---

hyp_to_rmd *Export hyp object to rmarkdown*

Description

Export hyp object to rmarkdown

Usage

```
hyp_to_rmd(
  hyp_obj,
  file_path,
  title = "Workflow Report",
  subtitle = "",
  author = "",
  header = "Results",
  versioning = TRUE,
  show_dots = TRUE,
  show_emaps = TRUE,
  show_hmaps = FALSE,
  show_tables = TRUE,
  hyp_dots_args = list(top = 15, val = "fdr"),
 hyp_emap_args = list(top = 25, val = "fdr", similarity_metric = "jaccard_similarity",
    similarity_cutoff = 0.2),
  hyp_hmap_args = list(top = 25, val = "fdr"),
  custom_rmd_config = NULL,
  custom_pre_content = NULL,
  custom_post_content = NULL,
  session_info = FALSE
)
```

Arguments

hyp_obj	A hyp object, multihyp object, or list of multihyp objects
file_path	A file path
title	Title of markdown report
subtitle	Subtitle of markdown report
author	Authors of markdown report
header	Header name of tabset section
versioning	Add versioning information
show_dots	Option to show dots plots in tabs
show_emaps	Option to show enrichment maps in tabs
show_hmaps	Option to show hiearchy maps in tabs
show_tables	Option to show table in tabs
hyp_dots_args	A list of keyword arguments passed to hyp_dots
hyp_emap_args	A list of keyword arguments passed to hyp_emap
hyp_hmap_args	A list of keyword arguments passed to hyp_hmap
custom_rmd_config	Replace configuration section of markdown report
custom_pre_content	Insert custom content before tabset section
custom_post_content	Insert custom content after tabset section
session_info	Use true to include session info

Value

NULL

Examples

```
genesets <- list("GSET1" = c("GENE1","GENE2","GENE3"),
                 "GSET2" = c("GENE4","GENE5","GENE6"))

signature <- c("GENE1","GENE2","GENE7")

hyp_obj <- hypeR(signature, genesets, background=20)

rmd_path <- file.path(tempdir(), "report.rmd")
hyp_to_rmd(hyp_obj,
          file_path=rmd_path,
          title="Example Report",
          show_dots=FALSE,
          show_emaps=FALSE,
          show_hmaps=FALSE,
          show_tables=TRUE,
          versioning=FALSE)
```

---

hyp_to_table *Export hyp/multihyp object to table*

Description

Export hyp/multihyp object to table

Usage

```
hyp_to_table(hyp_obj, file_path, sep = "\t", cols = NULL, versioning = TRUE)
```

Arguments

hyp_obj	A hyp or multihyp object
file_path	A file path for hyp objects and directory for multihyp objects
sep	The field separator string
cols	Dataframe columns to include
versioning	Add header with versioning information

Examples

```
genesets <- msigdb_gsets("Homo sapiens", "C2", "CP:KEGG_LEGACY")

signature <- c("IDH3B","DLST","PCK2","CS","PDHB","PCK1","PDHA1","LOC642502",
               "PDHA2","LOC283398","FH","SDHD","OGDH","SDHB","IDH3A","SDHC",
               "IDH2","IDH1","OGDHL","PC","SDHA","SUCLG1","SUCLA2","SUCLG2")

hyp_obj <- hypeR(signature, genesets, background=2522)

hyp_to_table(hyp_obj, file_path="pathways.txt")
```

---

hypeR *Calculate enrichment of one or more signatures*

Description

Calculate enrichment of one or more signatures

Usage

```
hypeR(
  signature,
  genesets,
  test = c("hypergeometric", "kstest"),
  background = 23467,
  power = 1,
  absolute = FALSE,
  pval = 1,
  fdr = 1,
  plotting = FALSE,
  quiet = TRUE
)
```

Arguments

signature	A vector of symbols
genesets	A gsets/rgsets object or a named list of genesets
test	Choose an enrichment type e.g. c("hypergeometric", "kstest")
background	Size or character vector of background population genes
power	Exponent for weights (kstest only)
absolute	Takes max-min score rather than the max deviation from null (kstest only)
pval	Filter results to be less than pval cutoff
fdr	Filter results to be less than fdr cutoff
plotting	Use true to generate plots for each geneset test (may slow performance)
quiet	Use true to suppress logs and warnings

Value

A hyp object

Examples

```
genesets <- msigdb_gsets("Homo sapiens", "C2", "CP:KEGG_LEGACY")

signature <- c("IDH3B","DLST","PCK2","CS","PDHB","PCK1","PDHA1","LOC642502",
               "PDHA2","LOC283398","FH","SDHD","OGDH","SDHB","IDH3A","SDHC",
               "IDH2","IDH1","OGDHL","PC","SDHA","SUCLG1","SUCLA2","SUCLG2")

hyp_obj <- hypeR(signature, genesets, background=2522)
```

---

hyperdb_available *Check available data to download from hyperdb*

Description

Check available data to download from hyperdb

Usage

```
hyperdb_available()
```

Examples

```
hyperdb_available()
```

---

hyperdb_gsets *Download data from hyperdb*

Description

Download data from hyperdb

Usage

```
hyperdb_gsets(source, gsets)
```

Arguments

source	A source identifier
gsets	A genesets identifier

Value

A list

Examples

```
KEGG <- hyperdb_gsets("KEGG", "KEGG_v92.0.rds")
```

---

hyperdb_rgsets *Download data from hyperdb in the form of a rgsets object*

Description

Download data from hyperdb in the form of a rgsets object

Usage

```
hyperdb_rgsets(rgsets, version)
```

Arguments

rgsets	A name corresponding to an available relational genesets object
version	A version number

Value

An rgsets object

Examples

```
REACTOME <- hyperdb_rgsets("REACTOME", "70.0")
```

---

limma *Differential Expression*

Description

A differential expression table

Usage

```
limma
```

Format

A data frame

msigdb_available *Get msigdbr available genesets*

Description

Get msigdbr available genesets

Usage

```
msigdb_available(species = "Homo sapiens")
```

Arguments

species	A species to determine gene symbols (refer to ?msigdbr::msigdbr for available species)

Value

A dataframe of available genesets

Examples

```
msigdb_available("Homo sapiens")
```

msigdb_download *Download data from msigdb in the form of a named list*

Description

Download data from msigdb in the form of a named list

Usage

```
msigdb_download(species, collection, subcollection = NULL)
```

Arguments

species	A species to determine gene symbols (refer to ?msigdbr::msigdbr for available species)
collection	Geneset collection (refer to ?msigdbr::msigdbr_collections for available categories)
subcollection	Geneset subcollection (refer to ?msigdbr::msigdbr_collections for available subcategories)

Value

A list of genesets

Examples

```
HALLMARK_HUMAN <- msigdb_download("Homo sapiens", "H")
HALLMARK_MOUSE <- msigdb_download("Mus musculus", "MH")
```

`msigdb_gsets` *Download data from msigdb in the form of a gsets object*

Description

Download data from msigdb in the form of a gsets object

Usage

```
msigdb_gsets(species, collection, subcollection = NULL, clean = FALSE)
```

Arguments

`species`	A species to determine gene symbols (refer to ?msigdbr::msigdbr_species for available species)
`collection`	Geneset collection (refer to ?msigdbr::msigdbr_collections for available categories)
`subcollection`	Geneset subcollection (refer to ?msigdbr::msigdbr_collections for available subcategories)
`clean`	Use true to clean labels of genesets

Value

A gsets object

Examples

```
HALLMARK_HUMAN <- msigdb_gsets("Homo sapiens", "H")
HALLMARK_MOUSE <- msigdb_gsets("Mus musculus", "MH")
```

msigdb_info *Print msigdb gsets information*

Description

Print msigdb gsets information

Usage

```
msigdb_info()
```

Examples

```
msigdb_info()
```

msigdb_species *Get msigdbr available species*

Description

Get msigdbr available species

Usage

```
msigdb_species()
```

Value

A character vector of species

Examples

```
msigdb_species()
```

msigdb_version *Get msigdbr package version number*

Description

Get msigdbr package version number

Usage

```
msigdb_version()
```

Value

Version number

Examples

```
msigdb_version()
```

multihyp *A multihyp object*

Description

A multihyp object

A multihyp object

See Also

hyp

Public fields

`data` A list of hyp objects

Methods

Public methods:

- `multihyp$new()`
- `multihyp$print()`
- `multihyp$as.list()`
- `multihyp$clone()`

Method `new()`**:** Create a multihyp object

Usage:

```
multihyp$new(data)
```

Arguments:

`data` A list of hyp objects

Returns: A new multihyp object

Method `print()`**:** Print multihyp obect

Usage:

```
multihyp$print()
```

Returns: NULL

Method `as.list()`**:** Print multihyp obect

Usage:

```
multihyp$as.list()
```

Returns: A list of hyp objects as dataframes

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
multihyp$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
data <- data.frame(replicate(5,sample(0:1,10,rep=TRUE)))
args <- list("arg_1"=1, "arg_2"=2, "arg_3"=3)
hyp_obj <- hyp$new(data, args=args)
data <- list("hyp_1"=hyp_obj, "hyp_2"=hyp_obj,"hyp_3"=hyp_obj)
multihyp_obj <- multihyp$new(data)
```

`rctbl_build` *Reactable builder for hyp or mhyp objects*

Description

Reactable builder for hyp or mhyp objects

Usage

```
rctbl_build(obj, ...)
```

Arguments

`obj`	A hyp or multihyp object
`...`	Arguments passed to table generators

Examples

```
genesets <- msigdb_gsets("Homo sapiens", "C2", "CP:KEGG_LEGACY")$genesets[1:5]

experiment <- list("S1"=c("IDH3B","DLST","PCK2","CS","PDHB","PCK1","PDHA1","LOC642502"),
                   "S2"=c("PDHA2","LOC283398","FH","SDHD","OGDH","SDHB","IDH3A","SDHC"))

mhyp_obj <- hypeR(experiment, genesets, background=2522)

rctbl_build(mhyp_obj)
```

---

rctbl_hyp *Reactable table for hyp objects*

Description

Reactable table for hyp objects

Usage

```
rctbl_hyp(
  hyp,
  type = c("inner", "outer"),
  show_emaps = FALSE,
  show_hmaps = FALSE,
  hyp_emap_args = list(top = 25, val = "fdr"),
  hyp_hmap_args = list(top = 25, val = "fdr")
)
```

Arguments

hyp	A hyp object
type	Use style class for outer or inner tables
show_emaps	Option to show enrichment maps in tabs
show_hmaps	Option to show hiearchy maps in tabs
hyp_emap_args	A list of keyword arguments passed to hyp_emap
hyp_hmap_args	A list of keyword arguments passed to hyp_hmap

Examples

```
genesets <- msigdb_gsets("Homo sapiens", "C2", "CP:KEGG_LEGACY")$genesets[1:5]

signature <- c("IDH3B","DLST","PCK2","CS","PDHB","PCK1","PDHA1","LOC642502",
               "PDHA2","LOC283398","FH","SDHD","OGDH","SDHB","IDH3A","SDHC",
               "IDH2","IDH1","OGDHL","PC","SDHA","SUCLG1","SUCLA2","SUCLG2")

hyp_obj <- hypeR(signature, genesets, background=2522)
```

```
rctbl_hyp(hyp_obj)
```

`rctbl_mhyp` *Reactable table for multihyp objects*

Description

Reactable table for multihyp objects

Usage

```
rctbl_mhyp(
  mhyp,
  show_emaps = FALSE,
  show_hmaps = FALSE,
  hyp_emap_args = list(top = 25, val = "fdr"),
  hyp_hmap_args = list(top = 25, val = "fdr")
)
```

Arguments

`mhyp`	A multihyp object
`show_emaps`	Option to show enrichment maps in tabs
`show_hmaps`	Option to show hiearchy maps in tabs
`hyp_emap_args`	A list of keyword arguments passed to hyp_emap
`hyp_hmap_args`	A list of keyword arguments passed to hyp_hmap

Examples

```
genesets <- msigdb_gsets("Homo sapiens", "C2", "CP:KEGG_LEGACY")$genesets[1:5]

experiment <- list("S1"=c("IDH3B","DLST","PCK2","CS","PDHB","PCK1","PDHA1","LOC642502"),
                   "S2"=c("PDHA2","LOC283398","FH","SDHD","OGDH","SDHB","IDH3A","SDHC"))

mhyp_obj <- hypeR(experiment, genesets, background=2522)

rctbl_mhyp(mhyp_obj)
```

rgsets *A relational genesets object*

Description

A relational genesets object

A relational genesets object

See Also

gsets

Public fields

genesets A list of genesets where list names refers to geneset labels and values are geneset members represented as a vector

nodes A data frame of labeled nodes

edges A data frame of directed edges

name A character vector describing source of genesets

version A character vector describing versioning

Methods

Public methods:

- rgsets$new()
- rgsets$print()
- rgsets$info()
- rgsets$reduce()
- rgsets$subset()
- rgsets$clone()

Method new(): Create a rgsets object

Usage:

```
rgsets$new(
  genesets,
  nodes,
  edges,
  name = "Custom",
  version = "",
  quiet = FALSE
)
```

Arguments:

genesets A list of genesets where list names refers to geneset labels and values are geneset members represented as a vector

`nodes` A data frame of labeled nodes

`edges` A data frame of directed edges

`name` A character vector describing source of genesets

`version` A character vector describing versioning

`quiet` Use true to silence warnings

Returns: A new rgsets object

Method `print()`**:** Print relational genesets information

Usage:

```
rgsets$print()
```

Returns: NULL

Method `info()`**:** Returns versioning information

Usage:

```
rgsets$info()
```

Returns: A character vector with name and version

Method `reduce()`**:** Reduces genesets to a background distribution of symbols

Usage:

```
rgsets$reduce(background)
```

Arguments:

`background` A character vector of symbols

Returns: A rgsets object

Method `subset()`**:** Subsets genesets on a character vector of labels

Usage:

```
rgsets$subset(labels)
```

Arguments:

`labels` A character vector of genesets

Returns: A rgsets object

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
rgsets$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
genesets <- list("GSET1" = c("GENE1","GENE2","GENE3"),
                 "GSET2" = c("GENE4","GENE5","GENE6"))

nodes <- data.frame(label=c("GSET1","GSET2","PARENT"),
                    row.names=c("N1","N2","N3"))

edges <- data.frame(from=c("N3","N3"), to=c("N1","N2"))

rgsets_obj <- rgsets$new(genesets, nodes, edges, name="example", version="v1.0")
print(rgsets_obj)
```

---

wgcna *Co-expression Modules*

Description

A list of co-expression modules

Usage

```
wgcna
```

Format

A nested list of character vectors

Index

∗ **datasets**
    limma, 21
    wgcna, 31

clean_genesets, 3

enrichr_available, 3
enrichr_download, 4
enrichr_gsets, 4

genesets_Server, 5
genesets_UI, 6
ggempty, 6
ggeplot, 7
ggvenn, 7
gsets, 8

hyp, 10
hyp_dots, 11
hyp_emap, 12
hyp_hmap, 13
hyp_show, 14
hyp_to_excel, 15
hyp_to_graph, 16
hyp_to_rmd, 16
hyp_to_table, 18
hypeR, 19
hyperdb_available, 20
hyperdb_gsets, 20
hyperdb_rgsets, 21

limma, 21

msigdb_available, 22
msigdb_download, 22
msigdb_gsets, 23
msigdb_info, 24
msigdb_species, 24
msigdb_version, 25
multihyp, 25

rctbl_build, 26
rctbl_hyp, 27
rctbl_mhyp, 28
rgsets, 29

wgcna, 31