

Package ‘hypeR’

October 17, 2020

Title Hyper Enrichment

Version 1.4.0

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 7.0.2

LazyData true

Imports ggplot2, ggforce, R6, magrittr, dplyr, purrr, stats, stringr,
scales, rlang, httr, openxlsx, reactable, msigdb, kableExtra,
rmarkdown, igraph, visNetwork

Suggests devtools, testthat, knitr

biocViews GeneSetEnrichment, Annotation, Pathways

VignetteBuilder knitr

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

git_url <https://git.bioconductor.org/packages/hypeR>

git_branch RELEASE_3_11

git_last_commit 21e0dc0

git_last_commit_date 2020-04-27

Date/Publication 2020-10-16

Author Anthony Federico [aut, cre],
Stefano Monti [aut]

Maintainer Anthony Federico <anfed@bu.edu>

R topics documented:

enrichr_available	2
enrichr_download	3
enrichr_gsets	3
ggempty	4
ggeplot	4

ggvenn	5
gsets	5
hyp	6
hypeR	8
hyperdb_info	9
hyperdb_rgsets	9
hyp_dots	10
hyp_emap	11
hyp_hmap	12
hyp_show	13
hyp_to_excel	13
hyp_to_rmd	14
hyp_to_table	15
msigdb_available	16
msigdb_download	16
msigdb_gsets	17
msigdb_info	17
msigdb_species	18
msigdb_version	18
multihyp	19
pvector	20
rgsets	21
Index	23

enrichr_available *Get enrichr available genesets*

Description

Get enrichr available genesets

Usage

```
enrichr_available()
```

Value

A dataframe of available genesets

Examples

```
enrichr_available()
```

enrichr_download	<i>Download data from enrichr in the form of a named list</i>
------------------	---

Description

Download data from enrichr in the form of a named list

Usage

```
enrichr_download(genesets)
```

Arguments

genesets A name corresponding to available genesets

Value

A list of genesets

Examples

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

enrichr_gsets	<i>Download data from enrichr in the form of a gsets object</i>
---------------	---

Description

Download data from enrichr in the form of a gsets object

Usage

```
enrichr_gsets(genesets)
```

Arguments

genesets A name corresponding to available genesets

Value

A gsets object

Examples

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

`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

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

<code>a</code>	A vector for group a
<code>b</code>	A vector for group b
<code>ga</code>	A string label for group a
<code>gb</code>	A string label for group b
<code>title</code>	Plot title

Value

A ggplot object

`gsets`*A genesets object*

Description

A genesets object

A genesets object

Arguments

genesets A named list of genesets
name A character vector describing source of genesets
version A character vector describing versioning

Methods

`print(gsets)` prints geneset information.

`gsets$reduce(background)` reduces genesets to a background distribution of symbols.

See Also

`rgsets`

Methods

Public methods:

- `gsets$new()`
- `gsets$print()`
- `gsets$reduce()`
- `gsets$clone()`

Method `new()`:

Usage:

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

Method `print()`:

Usage:

```
gsets$print(...)
```

Method `reduce()`:

Usage:

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

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

Arguments

data A dataframe returned by `hyper`

plots A list of plots returned by `hyper`

args A list of arguments passed to `hyper`

Methods

`print(hyp)` shows some information about the object data as well as the arguments used in creating it.

`hyp$as.data.frame()` returns the dataframe slot.

See Also

`multihyp`

Methods

Public methods:

- [hyp\\$new\(\)](#)
- [hyp\\$print\(\)](#)
- [hyp\\$as.data.frame\(\)](#)
- [hyp\\$clone\(\)](#)

Method `new()`:

Usage:

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

Method `print()`:

Usage:

```
hyp$print(...)
```

Method `as.data.frame()`:

Usage:

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

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)
```

hyperR

*Calculate enrichment of one or more signatures***Description**

Calculate enrichment of one or more signatures

Usage

```
hyperR(
  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")

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 <- hyperR(signature, genesets, background=2522)
```

hyperdb_info	<i>Print hypeR-db rgsets information</i>
--------------	--

Description

Print hypeR-db rgsets information

Usage

```
hyperdb_info()
```

Examples

```
hyperdb_info()
```

hyperdb_rgsets	<i>Download data from hyperdb in the form of a rgsets object</i>
----------------	--

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")
```

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,
  sizes = TRUE,
  pval = 1,
  fdr = 1,
  val = c("fdr", "pval"),
  title = ""
)
```

Arguments

hyp_obj	A hyp or multihyp object
top	Limit number of genesets shown
abrv	Abbreviation length of geneset labels
sizes	Size dots by geneset sizes
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

Value

A ggplot object

Examples

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

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")

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 hierarchy map*

Description

Visualize hyp/multihyp objects as a hierarchy map

Usage

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

Arguments

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

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")

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, version = TRUE)
```

Arguments

hyp_obj	A hyp or multihyp object
file_path	A file path
cols	Dataframe columns to include
version	Add sheet with versioning information

Examples

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

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_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",
  version = 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
)
```

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

version	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 hierarchy 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

hyp_to_table	<i>Export hyp/multihyp object to table</i>
--------------	--

Description

Export hyp/multihyp object to table

Usage

```
hyp_to_table(hyp_obj, file_path, sep = "\t", cols = NULL, version = 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
version	Add header with versioning information

Examples

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

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")
```

msigdb_available *Get msigdb available genesets*

Description

Get msigdb available genesets

Usage

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

Arguments

species A species to determine gene symbols (refer to ?msigdb::msigdb 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, category, subcategory = "")
```

Arguments

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

category Geneset category (refer to ?msigdb::msigdb for available categories)

subcategory Geneset subcategory (refer to ?msigdb::msigdb for available subcategories)

Value

A list of genesets

Examples

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

msigdb_gsets	<i>Download data from msigdb in the form of a gsets object</i>
--------------	--

Description

Download data from msigdb in the form of a gsets object

Usage

```
msigdb_gsets(species, category, subcategory = "")
```

Arguments

species	A species to determine gene symbols (refer to ?msigdb::msigdb for available species)
category	Geneset category (refer to ?msigdb::msigdb for available categories)
subcategory	Geneset subcategory (refer to ?msigdb::msigdb for available subcategories)

Value

A gsets object

Examples

```
HALLMARK <- msigdb_gsets("Homo sapiens", "H", "")
```

msigdb_info	<i>Print msigdb gsets information</i>
-------------	---------------------------------------

Description

Print msigdb gsets information

Usage

```
msigdb_info()
```

Examples

```
msigdb_info()
```

msigdb_species	<i>Get msigdb available species</i>
----------------	-------------------------------------

Description

Get msigdb available species

Usage

```
msigdb_species()
```

Value

A character vector of species

Examples

```
msigdb_species()
```

msigdb_version	<i>Get msigdb package version number</i>
----------------	--

Description

Get msigdb package version number

Usage

```
msigdb_version()
```

Value

Version number

Examples

```
msigdb_version()
```

multihyp

A multihyp object

Description

A multihyp object

A multihyp object

Arguments

data A list of hyp objects

Methods

`print(multihyp)` shows some information about the object data

`multihyp$as.list()` returns a list of hyp objects as dataframes.

See Also

hyp

Methods

Public methods:

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

Method `new()`:

Usage:

`multihyp$new(data)`

Method `print()`:

Usage:

`multihyp$print(...)`

Method `as.list()`:

Usage:

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

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)
```

pvector

A push/pop capable vector

Description

A push/pop capable vector

A push/pop capable vector

Arguments

values A vector of values

Methods

`print(pvector)` prints vector values.

`pvector$length()` returns length of vector values.

`pvector$pop()` pops vector, returning popped value.

`pvector$push()` pushes to vector.

Methods

Public methods:

- `pvector$new()`
- `pvector$print()`
- `pvector$length()`
- `pvector$pop()`
- `pvector$push()`
- `pvector$clone()`

Method `new()`:

Usage:

`pvector$new(values = c())`

Method `print()`:

Usage:

`pvector$print(...)`

Method `length()`:

Usage:

`pvector$length()`

Method pop():*Usage:*

pvector\$pop()

Method push():*Usage:*

pvector\$push(pushed.values)

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

pvector\$clone(deep = FALSE)

Arguments:

deep Whether to make a deep clone.

Examples

```
pv <- pvector$new(c(1,2,3))
popped <- pv$pop()
pv$push(4)
pv$push(c(5,6))
print(pv)
```

 rgsets

A relational genesets object

Description

A relational genesets object

A relational genesets object

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 e.g.

	label
G1	Geneset 1
G2	Geneset 2
G3	Geneset 3

edges A data frame of directed edges

from	to
G1	G2

G1 G3

name A character vector describing source of genesets

version A character vector describing versioning

Methods

`print(rgsets)` shows some information about the object data

`rgsets$subset(labels)` returns an `rgsets` object subsetted on geneset labels

See Also

`gsets` `pvector`

Methods**Public methods:**

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

Method `new()`:

Usage:

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

Method `print()`:

Usage:

```
rgsets$print(...)
```

Method `reduce()`:

Usage:

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

Method `subset()`:

Usage:

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

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

```
testdat <- readRDS(file.path(system.file("extdata", package="hyper"), "testdat.rds"))
rgsets <- rgsets$new(genesets=testdat$genesets, nodes=testdat$nodes, edges=testdat$edges, name="Example", version="1.0")
```

Index

enrichr_available, 2
enrichr_download, 3
enrichr_gsets, 3

ggempty, 4
ggeplot, 4
ggvenn, 5
gsets, 5

hyp, 6
hyp_dots, 10
hyp_emap, 11
hyp_hmap, 12
hyp_show, 13
hyp_to_excel, 13
hyp_to_rmd, 14
hyp_to_table, 15
hyper, 8
hyperdb_info, 9
hyperdb_rgsets, 9

msigdb_available, 16
msigdb_download, 16
msigdb_gsets, 17
msigdb_info, 17
msigdb_species, 18
msigdb_version, 18
multihyp, 19

pvector, 20

rgsets, 21