Package ‘yulab.utils’

November 7, 2024

Title Supporting Functions for Packages Maintained by 'YuLab-SMU'

Version 0.1.8

Description Miscellaneous functions commonly used by 'YuLab-SMU'.

Imports cli, digest, fs, rlang, tools, utils

Suggests httr2, jsonlite, openssl, rappdirs

ByteCompile true

License Artistic-2.0

URL `https://yulab-smu.top/`

BugReports `https://github.com/YuLab-SMU/yulab.utils/issues`

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Guangchuang Yu [aut, cre] (<`https://orcid.org/0000-0002-6485-8781`>)

Maintainer Guangchuang Yu <`guangchuangyu@gmail.com`>

Repository CRAN

Date/Publication 2024-11-07 19:50:02 UTC

Contents

`check_pkg` *check_pkg*

Description

Check whether the input packages are installed

Usage

```
check_pkg(pkg, reason = NULL, ...)
```

Arguments

`pkg`	package names
`reason`	the reason to check the pkg. If NULL, it will set the reason to the parent call.
`...`	additional parameters that passed to `rlang::check_installed()`

Details

This function check whether the input packages are installed. If not, it asks the user whether to install the missing packages.

Value

see also check_installed

Author(s)

Guangchuang Yu

`combinations` *combinations*

Description

all possible combinations of n sets

Usage

```
combinations(n)
```

Arguments

`n`	number of sets

Value

a list of all combinations

`CRANpkg` *print md text of package with link to homepage (CRAN or Bioconductor)*

Description

print md text of package with link to homepage (CRAN or Bioconductor)

Usage

```
CRANpkg(pkg)

Biocpkg(pkg)
```

Arguments

`pkg`	package name

Value

md text string

Author(s)

Guangchuang Yu

`exec` *exec*

Description

run system command

Usage

```
exec(command)
```

Arguments

`command`	system command to run

Value

An `exec` instance that stores system command outputs

Author(s)

Guangchuang Yu

`get_dependencies` *get_dependencies*

Description

get reverse dependencies

Usage

```
get_dependencies(pkg, repo = c("CRAN", "BioC"))
```

Arguments

`pkg`	package name
`repo`	’CRAN’ and/or ’BioC’

Value

reverse dependencies

Author(s)

Guangchuang Yu

`get_fun_from_pkg` *get_fun_from_pkg*

Description

load function from package

Usage

```
get_fun_from_pkg(pkg, fun)
```

Arguments

`pkg`	package
`fun`	function

Value

function

Author(s)

Guangchuang Yu

Examples

```
get_fun_from_pkg('utils', 'zip')
```

`Githubpkg` *print md text of package with link to github repo*

Description

print md text of package with link to github repo

Usage

```
Githubpkg(user, pkg)
```

Arguments

`user`	github user
`pkg`	package name

Value

md text string

Author(s)

Guangchuang Yu

`has_internet` *has_internet*

Description

test for internect connection via reading lines from a URL

Usage

```
has_internet(site = "https://www.baidu.com/")
```

Arguments

`site`	URL to test connection

Value

logical value

Author(s)

Guangchuang Yu

`initial_cache` *cache intermediate data*

Description

Yulab provides a set of utilities to cache intermediate data, including initialize the cached item, update cached item and rmove the cached item, etc.

Usage

```
initial_cache()

get_cache()

rm_cache()

initial_cache_item(item)

get_cache_item(item)

rm_cache_item(item)

update_cache_item(item, elements)

get_cache_element(item, elements)
```

Arguments

`item`	the name of the cached item
`elements`	elements to be cached in the item

Value

return the cache environment, item or selected elements, depends on the functions.

Examples

```
## Not run:
 slow_fib <- function(x) {
     if (x < 2) return(1)
     slow_fib(x-2) + slow_fib(x-1)
 }

 fast_fib <- function(x) {
     if (x < 2) return(1)
     res <- get_cache_element('fibonacci', as.character(x))
     if (!is.null(res)) {
         return(res)
     }
     res <- fast_fib(x-2) + fast_fib(x-1)
     e <- list()
     e[[as.character(x)]] <- res
     update_cache_item('fibonacci', e)
     return(res)
 }

 system.time(slow_fib(30))
 system.time(fast_fib(30))
```

```
## End(Not run)
```

`install_zip` *install_zip*

Description

install R package from zip file of source codes

Usage

```
install_zip(file, args = "--no-build-vignettes")
```

Arguments

`file`	zip file
`args`	argument to build package

Value

No return value, called for install R package from zip file of source codes

Author(s)

Guangchuang Yu

`install_zip_gh` *install_zip_gh*

Description

install github package

Usage

```
install_zip_gh(repo, ref = "HEAD", args = "--no-build-vignettes")
```

Arguments

`repo`	github repo
`ref`	github branch, default is HEAD, which means the default branch of the GitHub repo
`args`	argument to build package

Details

it download the zip file first and use `install_zip` to install it

Value

No return value, called for installing github package

Author(s)

Guangchuang Yu

`is.installed` *is.installed*

Description

Check whether the input packages are installed

Usage

```
is.installed(packages)
```

Arguments

`packages`	package names

Details

This function check whether the input packages are installed

Value

logical vector

Author(s)

Guangchuang Yu

Examples

```
is.installed(c("dplyr", "ggplot2"))
```

`ls2df` *Convert a list of vector (e.g, gene IDs) to a data.frame object*

Description

Convert a list of vector to a data.frame object.

Usage

```
ls2df(inputList)
```

Arguments

`inputList`	A list of vector

Value

a data.frame object.

`mat2df` *mat2df*

Description

convert a matrix to a tidy data frame (from wide to long format as described in the tidyverse concept)

Usage

```
mat2df(x)
```

Arguments

`x`	the input matrix

Value

a data.frame in long format with the 'value' column stores the original values and 'row' and 'col' columns stored in row and column index as in x

Author(s)

Guangchuang Yu

Examples

```
x <- matrix(1:15, nrow = 3)
mat2df(x)
```

`mat2list` *mat2list*

Description

convert a matrix to a list

Usage

```
mat2list(x)
```

Arguments

`x`	the input matrix

Value

a list that contains matrix columns as its elements

Examples

```
x <- matrix(1:15, nrow = 3)
mat2list(x)
```

`mypkg` *mypkg*

Description

print md text of link to a pakcage

Usage

```
mypkg(pkg, url)
```

Arguments

`pkg`	package name
`url`	package url

Value

md text string

Author(s)

Guangchuang Yu

`o` *o*

Description

open selected directory or file

Usage

```
o(file = ".")
```

Arguments

`file`	to be open; open working directory by default

Value

No return value, called for opening specific directory or file

Author(s)

Guangchuang Yu

Examples

```
## Not run:
## to open current working directory
o()

## End(Not run)
```

`packageTitle` *packageTitle*

Description

Extract package title

Usage

```
packageTitle(pkg, repo = "CRAN")
```

Arguments

`pkg`	package name
`repo`	'CRAN' and/or 'BioC'

Value

reverse dependencies

Author(s)

Guangchuang Yu

`pload` *pload*

Description

loading a package

Usage

```
pload(package, action = "auto")
```

Arguments

`package`	package name
`action`	function used to install package. If 'action = "auto"', it will try to use 'BiocManager::install()' if it is available.

Details

The function use 'library()' to load the package. If the package is not installed, the function will try to install it before loading it.

Value

the selected package loaded to the R session

Author(s)

Guangchuang Yu

`rbindlist` *rbindlist*

Description

rbind a list

Usage

```
rbindlist(x)
```

Arguments

`x`	a list that have similar elements that can be rbind to a data.frame

Value

data.frame

Author(s)

Guangchuang Yu

`read.cb` *read.cb*

Description

read clipboard

Usage

```
read.cb(reader = read.table, ...)
```

Arguments

`reader`	function to read the clipboard
`...`	parameters for the reader

Value

clipboard content, output type depends on the output of the reader

Author(s)

Guangchuang Yu

`scale_range` *scale-range*

Description

normalized data by range

Usage

```
scale_range(data)
```

Arguments

`data`	the input data.

Value

normalized data

Author(s)

Guangchuang Yu

`scihub_dl` *download publication via scihub*

Description

using scihub to download publication using doi

Usage

```
scihub_dl(doi, scihub = "sci-hub.tw", download = TRUE)
```

Arguments

`doi`	doi
`scihub`	scihub website
`download`	whether download the pdf file

Value

pdf url

Author(s)

Guangchuang Yu

`set_PCRE` *switch regular expression style (PCRE vs TRE)*

Description

The `set_regexpr_style()` allows user to specify which style to be used, while the `auto_set_regexpr_style()` automatically set the style depdending on the operating system (TRE for Windows and PCRE for other OSs (Linux and Mac)).

Usage

```
set_PCRE()

set_TRE()

use_perl()

set_regexpr_style(style)

auto_set_regexpr_style()
```

Arguments

`style`	one of 'PCRE' or 'TRE'

Details

`set_PCRE()` force to use PCRE style while `set_TRE()` force to use TRE.

Note that all these functions are not change the behavior of `gsub()` and `regexpr()`. The functions are just set a global option to store the user's choice of whether using `perl = TRUE`.

Users can access the option via `use_perl()` and pass the return value to `gusb()` or `regexpr()` to specify the style in use.

Value

logical value of whether use perl

Author(s)

Guangchuang Yu

References

https://stackoverflow.com/questions/47240375/regular-expressions-in-base-r-perl-true-vs-the-defaul

---

`show_in_excel` *show_in_excel*

Description

Open data frame in Excel. It can be used in pipe.

Usage

```
show_in_excel(.data)
```

Arguments

`.data`	a data frame to be open

Value

original .data

Author(s)

Guangchuang Yu

`str_detect` *str_detect*

Description

Detect the presentce/absence of a match

Usage

```
str_detect(string, pattern, negate = FALSE)
```

Arguments

`string`	input string
`pattern`	pattern to look for
`negate`	if TRUE, inverts the resulting boolen vector

Value

logical vector

Author(s)

Guangchuang Yu

`str_extract` *str_extract*

Description

Extract a substring using a pattern

Usage

```
str_extract(string, pattern)
```

Arguments

`string`	input string
`pattern`	a regular expression to describe the pattern to extracted from the 'string'

Value

substring

Author(s)

Guangchuang Yu

`str_starts` *str_starts*

Description

Detect the presence or absence of a pattern at the beginning or end of a string or string vector.

Usage

```
str_starts(string, pattern, negate = FALSE)

str_ends(string, pattern, negate = FALSE)
```

Arguments

`string`	input string
`pattern`	pattern with which the string starts or ends
`negate`	if TRUE, return non-matching elements

Value

a logical vector

Author(s)

Guangchuang Yu

`str_wrap` *str_wrap*

Description

wraping long string to multiple lines

Usage

```
str_wrap(string, width = getOption("width"))
```

Arguments

`string`	input string
`width`	the maximum number of characters before wrapping to a new line

Value

update strings with new line character inserted

Author(s)

Guangchuang Yu

`yread_tsv` *yread*

Description

read file with caching

Usage

```
yread_tsv(
  file,
  reader = utils::read.delim,
  params = list(),
  cache_dir = tempdir()
)

yread(file, reader = readLines, params = list(), cache_dir = NULL)
```

Arguments

`file`	a file or url
`reader`	a function to read the 'file_url'
`params`	a list of parameters that passed to the 'reader'
`cache_dir`	a folder to store cache files. If set to NULL will disable cache.

Details

This function read a file (local or url) and cache the content.

Value

the output of using the 'reader' to read the 'file_url' with parameters specified by the 'params'

Author(s)

Yonghe Xia and Guangchuang Yu

`yulab_msg` *yulab_msg*

Description

Messages for R package developed by YuLab

Usage

```
yulab_msg(pkgname = NULL, n = 1)
```

Arguments

`pkgname`	package name
`n`	number of citation messages

Value

package message

Author(s)

Guangchuang Yu

Index