

# Package ‘yulab.utils’

August 17, 2024

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

**Version** 0.1.6

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

**Imports** cli, digest, fs, httr2, rlang, stats, tools, utils

**Suggests** httr, jsonlite, openssl

**ByteCompile** true

**URL** <https://yulab-smu.top/>

**License** Artistic-2.0

**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** 2024-08-17 06:00:01 UTC

## Contents

check_pkg . . . . .	2
combinations . . . . .	3
CRANpkg . . . . .	3
exec . . . . .	4
get_dependencies . . . . .	4
get_fun_from_pkg . . . . .	5
Githubpkg . . . . .	5
initial_cache . . . . .	6
install_zip . . . . .	7
install_zip_gh . . . . .	8
is.installed . . . . .	8
ls2df . . . . .	9
mat2df . . . . .	10

mat2list . . . . .	10
mypkg . . . . .	11
o . . . . .	11
packageTitle . . . . .	12
pload . . . . .	13
rbindlist . . . . .	13
read.cb . . . . .	14
scale_range . . . . .	14
scihub_dl . . . . .	15
set_PCRE . . . . .	15
show_in_excel . . . . .	16
str_extract . . . . .	17
str_starts . . . . .	18
str_wrap . . . . .	18
yread_tsv . . . . .	19
yulab_msg . . . . .	20

## Index 21

---

check_pkg	<i>check_pkg</i>
-----------	------------------

---

### 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 <code>rlang::check_installed()</code>

### 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	<i>combinations</i>
--------------	---------------------

---

**Description**

all possible combinations of n sets

**Usage**

```
combinations(n)
```

**Arguments**

n                    number of sets

**Value**

a list of all combinations

---

CRANpkg	<i>print md text of package with link to homepage (CRAN or Bioconductor)</i>
---------	--

---

**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**

<code>pkg</code>	package
<code>fun</code>	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**

<code>user</code>	github user
<code>pkg</code>	package name

**Value**

md text string

**Author(s)**

Guangchuang Yu

---

initial_cache	<i>cache intermediate data</i>
---------------	--------------------------------

---

**Description**

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

---

<code>install_zip_gh</code>	<i>install_zip_gh</i>
-----------------------------	-----------------------

---

**Description**

install github package

**Usage**

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

**Arguments**

<code>repo</code>	github repo
<code>ref</code>	github branch, default is HEAD, which means the default branch of the GitHub repo
<code>args</code>	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

---

<code>is.installed</code>	<i>is.installed</i>
---------------------------	---------------------

---

**Description**

Check whether the input packages are installed

**Usage**

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

**Arguments**

<code>packages</code>	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	<i>mypkg</i>
-------	--------------

---

**Description**

print md text of link to a package

**Usage**

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

**Arguments**

pkg	package name
url	package url

**Value**

md text string

**Author(s)**

Guangchuang Yu

---

o	<i>o</i>
---	----------

---

**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)
```

---

<code>packageTitle</code>	<i>packageTitle</i>
---------------------------	---------------------

---

**Description**

Extract package title

**Usage**

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

**Arguments**

<code>pkg</code>	package name
<code>repo</code>	'CRAN' and/or 'BioC'

**Value**

reverse dependencies

**Author(s)**

Guangchuang Yu

---

pload	<i>pload</i>
-------	--------------

---

**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 'Bioc-Manager::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	<i>rbindlist</i>
-----------	------------------

---

**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	<i>download publication via scihub</i>
-----------	--

---

**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	<i>switch regular expression style (PCRE vs TRE)</i>
----------	--

---

**Description**

The `set_regexpr_style()` allows user to specify which style to be used, while the `auto_set_regexpr_style()` automatically set the style depending 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 gsub() 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-default>

---

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_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	<i>str_starts</i>
------------	-------------------

---

**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	<i>str_wrap</i>
----------	-----------------

---

**Description**

wrapping 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 and Erqiang Hu

---

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

auto\_set\_regexpr\_style (set\_PCRE), 15

Biocpkg (CRANpkg), 3

check\_installed, 2

check\_pkg, 2

combinations, 3

CRANpkg, 3

exec, 4

get\_cache (initial\_cache), 6

get\_cache\_element (initial\_cache), 6

get\_cache\_item (initial\_cache), 6

get\_dependencies, 4

get\_fun\_from\_pkg, 5

Githubpkg, 5

initial\_cache, 6

initial\_cache\_item (initial\_cache), 6

install\_zip, 7

install\_zip\_gh, 8

is\_installed, 8

ls2df, 9

mat2df, 10

mat2list, 10

mypkg, 11

o, 11

packageTitle, 12

pload, 13

rbindlist, 13

read.cb, 14

rm\_cache (initial\_cache), 6

rm\_cache\_item (initial\_cache), 6

scale\_range, 14

scihub\_dl, 15

set\_PCRE, 15

set\_regexpr\_style (set\_PCRE), 15

set\_TRE (set\_PCRE), 15

show\_in\_excel, 16

str\_ends (str\_starts), 18

str\_extract, 17

str\_starts, 18

str\_wrap, 18

update\_cache\_item (initial\_cache), 6

use\_perl (set\_PCRE), 15

yread (yread\_tsv), 19

yread\_tsv, 19

yulab\_msg, 20