

Package ‘pkgndep’

November 10, 2021

Type Package

Title Check the Heaviness of Package Dependencies

Version 1.1.0

Date 2021-11-09

Depends R (>= 3.5.0)

Imports ComplexHeatmap (>= 2.0.0), GetoptLong, utils, grid, hash, methods, BiocManager, brew

Suggests knitr, rmarkdown, svglite, callr, rjson

Description It checks the heaviness of a package's dependencies, i.e. number of additional packages that are required for each of the direct dependency packages. For each package listed in the ``Depends``, ``Imports``, ``LinkingTo``, ``Suggests`` and ``Enhances`` fields in the DESCRIPTION file, it checks how many additional packages it brings in. The summary of the dependency is visualized by a customized heatmap. It also additionally creates a website that contains global package dependency analysis for all packages on CRAN and Bioconductor (retrieved on 2021-10-28).

URL <https://github.com/jokergoo/pkgndep>

VignetteBuilder knitr

License MIT + file LICENSE

NeedsCompilation no

Author Zuguang Gu [aut, cre] (<<https://orcid.org/0000-0002-7395-8709>>)

Maintainer Zuguang Gu <z.gu@dkfz.de>

Repository CRAN

Date/Publication 2021-11-10 14:00:02 UTC

R topics documented:

child_dependency	2
downstream_dependency	3
gini_index	3
heaviness	4

heaviness_on_children	5
heaviness_on_downstream	5
html_report	6
loaded_packages	7
load_all_pkg_dep	7
load_pkg_db	8
load_pkg_stat_snapshot	9
open_website	9
parent_dependency	10
pkgndep	10
plot.pkgndep	11
print.pkgndep	12
reformat_db	13
required_dependency_packages	14
upstream_dependency	14

Index 16

child_dependency	<i>Get child dependency for a package</i>
------------------	-------------------------------------------

Description

Get child dependency for a package

Usage

```
child_dependency(package, fields = NULL)
```

Arguments

package	Package name.
fields	Which fields in DESCRIPTION? Values should be in Depends, Imports, LinkingTo, Suggests and Enhances.

Details

The dependency information is based on packages retrieved from CRAN/Bioconductor on 2021-10-28.

Value

A data frame with child packages as well as its heaviness on its child packages.

Examples

```
## Not run:
child_dependency("ComplexHeatmap")

## End(Not run)
```

downstream_dependency *Get downstream dependency for a package*

Description

Get downstream dependency for a package

Usage

```
downstream_dependency(package)
```

Arguments

package Package name

Details

The dependency information is based on packages retrieved from CRAN/Bioconductor on 2021-10-28.

Value

A data frame with all downstream packages.

Examples

```
downstream_dependency("ComplexHeatmap")
```

gini_index *Gini index*

Description

Gini index

Usage

```
gini_index(v)
```

Arguments

v A numeric vector.

Examples

```
x = readRDS(system.file("extdata", "ComplexHeatmap_dep.rds", package = "pkgndep"))
gini_index(x$heaviness[x$which_required] + 2)
```

heaviness	<i>Heaviness from parent packages</i>
-----------	---------------------------------------

Description

Heaviness from parent packages

Usage

```
heaviness(x, rel = FALSE, a = 10)
```

Arguments

x	An object returned by pkgndep .
rel	Whether to return the absolute measure or the relative measure.
a	A constant added for calculating the relative measure.

Details

The heaviness from a parent package is calculated as follows: If package B is in the "Depends"/"Imports"/"LinkingTo" fields of package A, which means, package B is necessary for package A, denote v_1 as the total numbers of packages required for package A, and v_2 as the total number of required packages if moving package B to "Suggests" (which means, now B is not necessary when for A). The absolute measure is simply $v_1 - v_2$ and relative measure is $(v_1 + a)/(v_2 + a)$.

In the second scenario, if B is in the "Suggests/Enhances" fields of package A, now v_2 is the total number of required packages if moving B to "Imports", the absolute measure is $v_2 - v_1$ and relative measure is $(v_2 + a)/(v_1 + a)$.

Value

A numeric vector.

Examples

```
x = readRDS(system.file("extdata", "ComplexHeatmap_dep.rds", package = "pkgndep"))
heaviness(x)
heaviness(x, rel = TRUE)
```

heaviness_on_children *Heaviness on all child packages*

Description

Heaviness on all child packages

Usage

```
heaviness_on_children(package, add_values_attr = FALSE)
```

Arguments

package A package name.
add_values_attr Whether to include "values" attribute? Internally used.

Value

The value is the mean heaviness of the package on all its child packages.

Examples

```
heaviness_on_children("ComplexHeatmap")
```

heaviness_on_downstream
Heaviness on all downstream packages

Description

Heaviness on all downstream packages

Usage

```
heaviness_on_downstream(package, move_to_suggests = TRUE, add_values_attr = FALSE)
```

Arguments

package A package name.
move_to_suggests Whether to move an Imports packages to Suggests or move a Suggests package to Imports?
add_values_attr Whether to include "values" attribute? Internally used.

Value

The value is the mean heaviness of the package on all its downstream packages. Denote n as the number of all its downstream packages, k_i as the number of required packages for package i (i.e. total packages loaded when only loading packages in `Depends`, `Imports` and `LinkingTo`), v_1 is the sum of required packages: $v_1 = \sum_i^n \{k_i\}$. Denote p_i as the number of required packages if moving package to `Suggests`, and v_2 as the sum of required packages: $v_2 = \sum_i^n \{p_i\}$. The final heaviness on downstream packages is $(v_1 - v_2)/n$.

Examples

```
## Not run:
heaviness_on_downstream("ComplexHeatmap")

## End(Not run)
```

html_report

HTML report for package dependency analysis

Description

HTML report for package dependency analysis

Usage

```
html_report(pkg, file = NULL)
```

Arguments

pkg	An object from pkgndep .
file	The path of the html file. If it is not specified, the report will be automatically opened in the web browser.

Value

The path of the HTML file of the report.

Example

```
if(interactive()) x = readRDS(system.file("extdata", "ComplexHeatmap_dep.rds", package = "pkgndep"))
html_report(x)
```

Examples

```
# There is no example
NULL
```

loaded_packages	<i>Loaded packages</i>
-----------------	------------------------

Description

Loaded packages

Usage

```
loaded_packages(pkg, verbose = TRUE)
```

Arguments

pkg	A package name.
verbose	Whether to print messages.

Details

It loads package pkg in a new R session and collects which other packages are loaded from [sessionInfo](#).

Value

A data frame.

Examples

```
loaded_packages("ComplexHeatmap")
```

load_all_pkg_dep	<i>Load all package dependency data</i>
------------------	-----------------------------------------

Description

Load all package dependency data

Usage

```
load_all_pkg_dep()
```

Details

It loads the package dependency analysis for all CRAN/Bioconductor packages done on 2021-10-28.

Value

A list of pkgndep objects where each element corresponds to the analysis on one package.

Examples

```
## Not run:
lt = load_all_pkg_dep()
length(lt)
head(names(lt))
lt[["ggplot2"]]

## End(Not run)
```

load_pkg_db

Load package database

Description

Load package database

Usage

```
load_pkg_db(lib = NULL, snapshot = FALSE, verbose = TRUE)
```

Arguments

lib	Local library path. If the value is NA, only remote package database is used.
snapshot	Internally used. If it is TRUE, the package database generated on 2021-10-28 is used.
verbose	Whether to print messages.

Details

It loads the package database from CRAN/Bioconductor and locally installed packages.

The database object internally is cached for repeated use of other functions in this package.

Value

A pkg_db class object.

Examples

```
## Not run:
pkg_db = load_pkg_db(lib = NA)
pkg_db

## End(Not run)
```

`load_pkg_stat_snapshot`*Load all package dependency statistics*

Description

Load all package dependency statistics

Usage

```
load_pkg_stat_snapshot()
```

Details

It loads the package dependency analysis for all CRAN/Bioconductor packages done on 2021-10-28.

Value

A data frame of various columns.

Examples

```
df = load_pkg_stat_snapshot()
head(df)
```

`open_website`*Open the online website of package dependency analysis*

Description

Open the online website of package dependency analysis

Usage

```
open_website()
```

Examples

```
# There is no example
NULL
```

parent_dependency *Get parent dependency for a package*

Description

Get parent dependency for a package

Usage

```
parent_dependency(package, fields = NULL)
```

Arguments

package	Package name.
fields	Which fields in DESCRIPTION? Values should be in Depends, Imports, LinkingTo, Suggests and Enhances.

Details

The dependency information is based on packages retrieved from CRAN/Bioconductor on 2021-10-28.

Value

A data frame with parent packages as well as their heaviness on package.

Examples

```
parent_dependency("ComplexHeatmap")
```

pkgndep *Number of dependency packages*

Description

Number of dependency packages

Usage

```
pkgndep(package, load = FALSE, verbose = TRUE)
```

Arguments

package	Package name if the package is already installed or the path of the package in the file system.
load	Whether also check which other packages are loaded into R session (directly or indirectly) when loading pkg.
verbose	Whether to show messages.

Details

The package database for dependency analysis is always directly retrieved from CRAN/Bioconductor.

Value

A pkgndep object.

Examples

```
## Not run:
x = pkgndep("ComplexHeatmap")

## End(Not run)
# The `x` variable generated by `pkgndep()` is already saved in this package.
x = readRDS(system.file("extdata", "ComplexHeatmap_dep.rds", package = "pkgndep"))
x
plot(x)
```

plot.pkgndep	<i>Make the dependency heatmap</i>
--------------	------------------------------------

Description

Make the dependency heatmap

Usage

```
## S3 method for class 'pkgndep'
plot(x, pkg_fontsize = 10*cex, title_fontsize = 12*cex,
      legend_fontsize = 10*cex, fix_size = !dev.interactive(), cex = 1,
      help = TRUE, file = NULL, res = 144, ...)
```

Arguments

x	An object from pkgndep .
pkg_fontsize	Fontsize for the package names.
title_fontsize	Fontsize for the title.
legend_fontsize	Fontsize for the legends.

<code>fix_size</code>	Should the rows and columns in the heatmap have fixed size?
<code>cex</code>	A factor multiplied to all font sizes.
<code>help</code>	Whether to print help message?
<code>file</code>	A path of the figure. The size of the figure is automatically calculated.
<code>res</code>	Resolution of the figure (only for png and jpeg).
<code>...</code>	Other arguments.

Details

If `fix_size` is set to `TRUE`. The size of the whole plot can be obtained by:

```
size = plot(x, fix_size = TRUE)
```

where `size` is a numeric vector of length two which are the width and height of the whole heatmap. If `file` argument is set, the size of the figure is automatically calculated.

If there are no dependency packages stored in `x`, `NULL` is returned.

Value

A vector of two numeric values (in inches) that correspond to the width and height of the plot.

Examples

```
# See examples in `pkgndep()`.
```

<code>print.pkgndep</code>	<i>Print method</i>
----------------------------	---------------------

Description

Print method

Usage

```
## S3 method for class 'pkgndep'
print(x, ...)
```

Arguments

<code>x</code>	An object from pkgndep .
<code>...</code>	Other arguments.

Value

No value is returned.

Examples

```
# See examples in `pkgndep()`.
```

reformat_db	<i>Format the package database</i>
-------------	------------------------------------

Description

Format the package database

Usage

```
reformat_db(db)
```

Arguments

db A data frame returned from [available.packages](#) or [installed.packages](#).

Details

It reformats the data frame of the package database into a pkg_db class object.

Value

A pkg_db class object. There are the following methods:

pkg_db\$get_meta(package, field=NULL) field can take values in "Package", "Version" and "Repository".

pkg_db\$get_dependency_table(package) Get the dependency table.

pkg_db\$get_rev_dependency_table(package) Get the reverse dependency table.

pkg_db\$package_dependencies(package, recursive=FALSE, reverse=FALSE, which="strong", simplify=FALSE)

All the arguments are the same as in [package_dependencies](#). Argument simplify controls whether to return a data frame or a simplified vector.

Examples

```
## Not run:
db = available.packages()
db2 = reformat_db(db)

## End(Not run)

# a pkg_db object generated on 2021-10-28 can be loaded by load_pkg_db()
db2 = load_pkg_db(snapshot = TRUE)
db2
db2$get_meta("ComplexHeatmap")
db2$get_dependency_table("ComplexHeatmap")
db2$get_rev_dependency_table("ComplexHeatmap")
db2$package_dependencies("ComplexHeatmap")
db2$package_dependencies("ComplexHeatmap", recursive = TRUE)
```

required_dependency_packages
Required dependency packages

Description

Required dependency packages

Usage

```
required_dependency_packages(x, all = FALSE)
```

Arguments

x	An object from pkgndep .
all	Whether to include the packages required if also including packages from "Suggests"/"Enhances" field.

Value

A vector of package names.

Examples

```
# There is no example  
NULL
```

upstream_dependency *Get upstream dependency for a package*

Description

Get upstream dependency for a package

Usage

```
upstream_dependency(package)
```

Arguments

package	Package name.
---------	---------------

Details

The dependency information is based on packages retrieved from CRAN/Bioconductor on 2021-10-28.

Value

A data frame with all upstream packages.

Examples

```
upstream_dependency("ComplexHeatmap")
```

Index

`available.packages`, [13](#)

`child_dependency`, [2](#)

`downstream_dependency`, [3](#)

`gini_index`, [3](#)

`heaviness`, [4](#)

`heaviness_on_children`, [5](#)

`heaviness_on_downstream`, [5](#)

`html_report`, [6](#)

`installed.packages`, [13](#)

`load_all_pkg_dep`, [7](#)

`load_pkg_db`, [8](#)

`load_pkg_stat_snapshot`, [9](#)

`loaded_packages`, [7](#)

`open_website`, [9](#)

`package_dependencies`, [13](#)

`parent_dependency`, [10](#)

`pkgndep`, [4](#), [6](#), [10](#), [11](#), [12](#), [14](#)

`plot.pkgndep`, [11](#)

`print.pkgndep`, [12](#)

`reformat_db`, [13](#)

`required_dependency_packages`, [14](#)

`sessionInfo`, [7](#)

`upstream_dependency`, [14](#)