

Package ‘BiocManager’

May 7, 2021

Title Access the Bioconductor Project Package Repository

Description A convenient tool to install and update Bioconductor packages.

Version 1.30.13

Imports utils

Suggests BiocVersion, remotes, rmarkdown, testthat, withr, curl, knitr

BugReports <https://github.com/Bioconductor/BiocManager/issues>

VignetteBuilder knitr

License Artistic-2.0

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Martin Morgan [aut] (<<https://orcid.org/0000-0002-5874-8148>>),
Marcel Ramos [ctb, cre] (<<https://orcid.org/0000-0002-3242-0582>>)

Maintainer Marcel Ramos <marcel.ramos@roswellpark.org>

Repository CRAN

Date/Publication 2021-05-07 14:50:02 UTC

R topics documented:

available	2
BiocManager-pkg	2
install	4
repositories	6
valid	7
version	9

Index	11
--------------	-----------

available *Discover packages available for installation.*

Description

Discover packages available for installation.

Usage

```
available(pattern = "", include_installed = TRUE)
```

Arguments

pattern character(1) pattern to filter (via `grep(pattern=...)`) available packages; the filter is not case sensitive.

include_installed logical(1) When TRUE, include installed packages in list of available packages; when FALSE, exclude installed packages.

Value

character() vector of package names available for installation.

Examples

```
avail <- BiocManager::available()
length(avail)

grep("bs.*hsapiens", avail, value = TRUE, ignore.case = TRUE)

## Not run:
  BiocManager::available("bs.*hsapiens")

## End(Not run)
```

BiocManager-pkg *Install or update Bioconductor, CRAN, or GitHub packages*

Description

This package provides tools for managing *Bioconductor* and other packages in a manner consistent with *Bioconductor*'s package versioning and release system.

Details

Main functions are as follows; additional help is available for each function, e.g., `?BiocManager::version`.

- `BiocManager::install()`: Install or update packages from *Bioconductor*, CRAN, and GitHub.
- `BiocManager::version()`: Report the version of *Bioconductor* in use.
- `BiocManager::available()`: Return a `character()` vector of package names available (at `BiocManager::repositories()`) for installation.
- `BiocManager::valid()`: Determine whether installed packages are from the same version of *Bioconductor*.
- `BiocManager::repositories()`: *Bioconductor* and other repository URLs to discover packages for installation.

The version of *Bioconductor* in use is determined by the installed version of a second package, `BiocVersion`. `BiocVersion` is installed automatically during first use of `BiocManager::install()`. If `BiocVersion` has not yet been installed, the version is determined by code in base R.

Options influencing package behavior (see `?options`, `?getOption`) include:

- `"repos"`, `"BiocManager.check_repositories"`, `"BiocManager.snapshot"`: URLs of additional repositories for use by `BiocManager::install()`. See `?repositories`.
- `"pkgType"`: The default type of packages to be downloaded and installed; see `?install.packages`.
- `"timeout"`: The maximum time allowed for download of a single package, in seconds. *BiocManager* increases this to 300 seconds to accommodate download of large BSgenome and other packages.

System environment variables influencing package behavior include:

- `BIOCONDUCTOR_ONLINE_VERSION_DIAGNOSIS` advanced configuration to avoid *Bioconductor* version checks. See `?install`.

Author(s)

Maintainer: Marcel Ramos <marcel.ramos@roswellpark.org> ([ORCID](#)) [contributor]

Authors:

- Martin Morgan <martin.morgan@roswellpark.org> ([ORCID](#))

See Also

Useful links:

- Report bugs at <https://github.com/Bioconductor/BiocManager/issues>

Examples

```
R.version.string
packageVersion("BiocManager")
if (requireNamespace("BiocVersion", quietly = TRUE))
  packageVersion("BiocVersion")
BiocManager::version()
```

install

Install or update Bioconductor, CRAN, and GitHub packages

Description

The `BiocManager::install()` function installs or updates *Bioconductor* and CRAN packages in a *Bioconductor* release. Upgrading to a new *Bioconductor* release may require additional steps; see <https://bioconductor.org/install>.

Usage

```
install(
  pkgs = character(),
  ...,
  site_repository = character(),
  update = TRUE,
  ask = TRUE,
  checkBuilt = FALSE,
  force = FALSE,
  version = BiocManager::version()
)
```

Arguments

<code>pkgs</code>	<code>character()</code> vector of package names to install or update. A missing value updates installed packages according to <code>update =</code> and <code>ask =</code> . Package names containing a <code>'/'</code> are treated as GitHub repositories and installed using <code>remotes::install_github()</code> .
<code>...</code>	Additional arguments used by <code>install.packages()</code> .
<code>site_repository</code>	(Optional) <code>character(1)</code> vector representing an additional repository in which to look for packages to install. This repository will be prepended to the default repositories (which you can see with <code>BiocManager::link{repositories}()</code>).
<code>update</code>	<code>logical(1)</code> . When <code>FALSE</code> , <code>BiocManager::install()</code> does not attempt to update old packages. When <code>TRUE</code> , update old packages according to <code>ask</code> .
<code>ask</code>	<code>logical(1)</code> indicating whether to prompt user before installed packages are updated. If <code>TRUE</code> , user can choose whether to update all outdated packages without further prompting, to pick packages to update, or to cancel updating (in a non-interactive session, no packages will be updated unless <code>ask = FALSE</code>).
<code>checkBuilt</code>	<code>logical(1)</code> . If <code>TRUE</code> a package built under an earlier major.minor version of R (e.g., 3.4) is considered to be old.
<code>force</code>	<code>logical(1)</code> . If <code>TRUE</code> re-download a package that is currently up-to-date.
<code>version</code>	<code>character(1)</code> <i>Bioconductor</i> version to install, e.g., <code>version = "3.8"</code> . The special symbol <code>version = "devel"</code> installs the current 'development' version.

Details

Installation of *Bioconductor* and CRAN packages use R's standard functions for library management – `install.packages()`, `available.packages()`, `update.packages()`. Installation of GitHub packages uses the `remotes::install_github()`.

When installing CRAN or *Bioconductor* packages, typical arguments include: `lib.loc`, passed to `\link{old.packages}()` and used to determine the library location of installed packages to be updated; and `lib`, passed to `\link{install.packages}()` to determine the library location where pkgs are to be installed.

When installing GitHub packages, `...` is passed to the **remotes** package functions `\link[remotes]{install_github}()` and `remotes::install()`. A typical use is to build vignettes, via `dependencies=TRUE`, `build_vignettes=TRUE`.

See `?repositories` for additional detail on customizing where BiocManager searches for package installation.

`BIOCONDUCTOR_ONLINE_VERSION_DIAGNOSIS` is an environment variable or global `options()` which, when set to `FALSE`, avoids the R and *Bioconductor* version checks that are done by querying an online configuration file. Setting `BIOCONDUCTOR_ONLINE_VERSION_DIAGNOSIS` to `FALSE` can speed package loading when internet access is slow or non-existent, but may result in out-of-date information about the current release and development versions of *Bioconductor*. Offline users should set the `BIOCONDUCTOR_CONFIG_FILE` environment variable or option to a `.yaml` file similar to <https://bioconductor.org/config.yaml> for full offline use and version validation.

Value

`BiocManager::install()` returns the `pkgs` argument, invisibly.

See Also

`BiocManager::\link{repositories}()` returns the *Bioconductor* and CRAN repositories used by `install()`. `\link{install.packages}()` installs the packages themselves (used by `BiocManager::install` internally).

`\link{update.packages}()` updates all installed packages (used by `BiocManager::install` internally).

`\link{chooseBioCmirror}()` allows choice of a mirror from all public *Bioconductor* mirrors.

`\link{chooseCRANmirror}()` allows choice of a mirror from all public CRAN mirrors.

Examples

```
## Not run:
## update previously installed packages
BiocManager::install()

## install Bioconductor packages, and prompt to update all
## installed packages
BiocManager::install(c("GenomicRanges", "edgeR"))

## install a CRAN and Bioconductor packages:
BiocManager::install(c("survival", "SummarizedExperiment"))

## install a package from source:
```

```
BiocManager::install("IRanges", type="source")

## End(Not run)
```

repositories

Display current Bioconductor and CRAN repositories.

Description

`repositories()` reports the URLs from which to install *Bioconductor* and CRAN packages. It is used by `BiocManager::install()` and other functions.

Usage

```
repositories(site_repository = character(), version = BiocManager::version())
```

Arguments

`site_repository`

(Optional) `character(1)` representing an additional repository (e.g., a URL to an organization's internally maintained repository) in which to look for packages to install. This repository will be prepended to the default repositories returned by the function.

`version`

(Optional) `character(1)` or `package_version` indicating the *Bioconductor* version (e.g., "3.8") for which repositories are required.

Details

`repositories()` returns the appropriate software package repositories for your version of *Bioconductor*.

Bioconductor has a 'release' and a 'devel' semi-annual release cycle. Packages within a release have been tested against each other and the current version of packages on CRAN. *Bioconductor* best practice is to use packages from the same release, and from the appropriate CRAN repository.

CRAN packages for out-of-date *Bioconductor* installations can be installed from historical 'snapshots' consistent with the last date the *Bioconductor* version was current. This behavior can be specified with `BiocManager.snapshot`. For example, *Bioconductor* version 3.11 was current until October 28, 2020; CRAN packages are therefore installed from a snapshot created on 2020-10-28. By default, the snapshots are from 'MRAN', the [Microsoft R Archive Network](#). Use `options(BiocManager.snapshot = "RSPM")` to instead use the [RStudio Package Manager](#), or `options(BiocManager.snapshot = "CRAN")` to use the current CRAN repository (i.e., disabling the snapshot feature).

It may be desirable to specify different default repositories, especially CRAN, for intentionally out-of-date *Bioconductor* releases (e.g., to support reproducible research). Use the approach provided by base *R* to specify alternative repositories, e.g., `options(repos = c(CRAN = "https://mran.microsoft.com/snapshot/"))`. This is supported, but generates an error because specification of an inappropriate CRAN repository (one providing packages not consistent with the dates of the *Bioconductor* release) results in use of CRAN packages not consistent with *Bioconductor* best practices.

If alternative default repositories are known to provide appropriate versions of CRAN or *Bioconductor* packages, the warning may be silenced (displayed as a message) with `options(BiocManager.check_repositories = FALSE)`. A message is still printed, to serve as a reminder when debugging problems related to incompatible package installation.

The intended use of `site_repository =` is to enable installation of packages not available in the default repositories, e.g., packages internal to an organization and not yet publicly available. A secondary use might provide alternative versions (e.g., compiled binaries) of packages available in the default repositories. Note that *R*'s standard rules of package selection apply, so the most recent version of candidate packages is selected independent of the location of the repository in the vector returned by `repositories()`.

For greater flexibility in installing packages while still adhering as much as possible to *Bioconductor* best practices, use `repositories()` as a basis for constructing the `repos =` argument to `install.packages()` and related functions.

Value

Named `character()` of repositories.

See Also

`BiocManager::\link{install}()` Installs or updates Bioconductor, CRAN, and GitHub packages.
`\link{chooseBioCmirror}()` choose an alternative Bioconductor mirror; not usually necessary.
`\link{chooseCRANmirror}()` choose an alternative CRAN mirror; not usually necessary.
`\link{setRepositories}()` Select additional repositories for searching.

Examples

```
BiocManager::repositories()
## Not run:
BiocManager::repositories(version="3.8")

## End(Not run)
```

valid

Validate installed package versions against correct versions.

Description

Check that installed packages are consistent (neither out-of-date nor too new) with the version of *R* and *Bioconductor* in use.

Usage

```
valid(
  pkgs = installed.packages(lib.loc, priority = priority),
  lib.loc = NULL,
  priority = "NA",
```

```

    type = getOption("pkgType"),
    filters = NULL,
    ...,
    checkBuilt = FALSE,
    site_repository = character()
)

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

```

Arguments

<code>pkgs</code>	A <code>character()</code> vector of package names for checking, or a matrix as returned by <code>\link{installed.packages}</code> .
<code>lib.loc</code>	A <code>character()</code> vector of library location(s) of packages to be validated; see <code>\link{installed.packages}()</code> .
<code>priority</code>	<code>character(1)</code> Check validity of all, "base", or "recommended" packages; see <code>\link{installed.packages}()</code> .
<code>type</code>	<code>character(1)</code> The type of available package (e.g., binary, source) to check validity against; see <code>\link{available.packages}()</code> .
<code>filters</code>	<code>character(1)</code> Filter available packages to check validity against; see <code>\link{available.packages}()</code> .
<code>...</code>	Additional arguments, passed to <code>BiocManager::\link{install}()</code> when <code>fix=TRUE</code> .
<code>checkBuilt</code>	<code>logical(1)</code> . If TRUE a package built under an earlier major.minor version of R (e.g., 3.4) is considered to be old.
<code>site_repository</code>	<code>character(1)</code> . See <code>?install</code> .
<code>x</code>	A 'biocValid' object returned by 'BiocManager::valid()'.

Details

This function compares the version of installed packages to the version of packages associated with the version of *R* and *Bioconductor* currently in use.

Packages are reported as 'out-of-date' if a more recent version is available at the repositories specified by `\BiocManager::repositories()`. Usually, `\BiocManager::install()` is sufficient to update packages to their most recent version.

Packages are reported as 'too new' if the installed version is more recent than the most recent available in the `\BiocManager::repositories()`. It is possible to down-grade by re-installing a too new package "PkgA" with `\BiocManger::install("PkgA")`. It is important for the user to understand how their installation became too new, and to avoid this in the future.

Value

biocValid list object with elements `too_new` and `out_of_date` containing `data.frames` with packages and their installed locations that are too new or out-of-date for the current version of *Bioconductor*. When internet access is unavailable, an empty 'biocValid' list is returned. If all packages ('pkgs') are up to date, then `TRUE` is returned.

'`print()`' is invoked for its side effect.

Author(s)

Martin Morgan <martin.morgan@roswellpark.org>

See Also

`BiocManager::link{install}()` to update installed packages.

Examples

```
BiocManager::valid()
```

version	<i>Version of Bioconductor currently in use.</i>
---------	--

Description

`version()` reports the version of *Bioconductor* appropriate for this version of R, or the version of *Bioconductor* requested by the user.

Usage

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

Arguments

x	An <code>unknown_version</code> instance used to represent the situation when the version of Bioconductor in use cannot be determined.
...	Additional arguments, ignored.

Details

`version()` (and all functions requiring version information) fails when version cannot be validated e.g., because internet access is not available.

Value

A two-digit version, e.g., 3.8, of class `package_version` describing the version of *Bioconductor* in use.

Examples

```
BiocManager::version()
```

Index

* **environment**

- install, [4](#)
- repositories, [6](#)
- valid, [7](#)
- `_PACKAGE (BiocManager-pkg)`, [2](#)
- available, [2](#)
- `BiocManager (BiocManager-pkg)`, [2](#)
- `BiocManager-pkg`, [2](#)
- `BiocManager.check_repositories (repositories)`, [6](#)
- `BiocManager.snapshot (repositories)`, [6](#)
- `BIOCONDUCTOR_ONLINE_VERSION_DIAGNOSIS (install)`, [4](#)
- install, [4](#)
- `print.biocValid (valid)`, [7](#)
- `print.version_sentinel (version)`, [9](#)
- repositories, [6](#)
- valid, [7](#)
- version, [9](#)