Package ‘BiocManager’

June 5, 2025

Title Access the Bioconductor Project Package Repository

Description A convenient tool to install and update Bioconductor packages.

Version 1.30.26

Imports utils

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

URL https://bioconductor.github.io/BiocManager/

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

VignetteBuilder knitr

License Artistic-2.0

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

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

Maintainer Marcel Ramos <marcel.ramos@sph.cuny.edu>

Repository CRAN

Date/Publication 2025-06-05 21:30:02 UTC

Contents

`BiocManager-package` *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"`: URLs of additional repositories for use by `BiocManger::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`.
- `BIOCONDUCTOR_CONFIG_FILE` for offline use of BiocManager versioning functionality. See `?install`.
- `BIOCONDUCTOR_USE_CONTAINER_REPOSITORY` opt out of binary package installations. See `?containerRepository`.
- `BIOCMANAGER_CHECK_REPOSITORIES` silence messages regarding non-standard CRAN or Bioconductor repositories. See `?repositories`.
- `BIOCMANAGER_SITE_REPOSITORY` configure a more permanent `site_repository` input to `repositories()`. See `?repositories`.
- `R_BIOC_VERSION` use a specific, possibly unsupported, version of Bioconductor. `install()`, `version()`, `available()`, `valid()`, and `repositories()` all use the version *without* checking that it is consistent with the version of *R* in use.

Author(s)

Maintainer: Marcel Ramos <`marcel.ramos@sph.cuny.edu`> (ORCID)

Authors:

- Martin Morgan <`martin.morgan@roswellpark.org`> (ORCID)

See Also

Useful links:

- `https://bioconductor.github.io/BiocManager/`
- Report bugs at `https://github.com/Bioconductor/BiocManager/issues`

Examples

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

`available` *Discover packages available for installation.*

Description

The function lists all packages available from `repositories()` when no pattern is provided. This usually includes CRAN and Bioconductor packages. The function can also be used to check for package name availability. Common use cases include annotation package lookups by organism short name (e.g., "hsapiens").

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

```
if (interactive()) {
  avail <- BiocManager::available()
  length(avail)

  BiocManager::available("bs.*hsapiens")
}
```

`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

`pkgs`	`character()` vector of package names to install or update. A missing value updates installed packages according to `update =` and `ask =`. Package names containing a '/' are treated as GitHub repositories and installed using `remotes::install_github()`.
`...`	Additional arguments used by `install.packages()`.
`site_repository`	(Optional) `character(1)` 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 `BiocManager::repositories()`).
`update`	`logical(1)`. When `FALSE`, `BiocManager::install()` does not attempt to update old packages. When `TRUE`, update old packages according to `ask`.
`ask`	`logical(1)` indicating whether to prompt user before installed packages are updated. If TRUE, 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 `ask = FALSE`).

`checkBuilt`	`logical(1)`. If `TRUE` a package built under an earlier major.minor version of R (e.g., 3.4) is considered to be old.
`force`	`logical(1)`. If `TRUE` re-download a package that is currently up-to-date.
`version`	`character(1)` *Bioconductor* version to install, e.g., `version = "3.8"`. The special symbol `version = "devel"` 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 `old.packages()` and used to determine the library location of installed packages to be updated; and `lib`, passed to `install.packages{}` to determine the library location where `pkgs` are to be installed.

When installing GitHub packages, `...` is passed to the **remotes** package functions `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`, allows organizations and its users to use offline repositories with BiocManager while enforcing appropriate version checks between *Bioconductor* and R. Setting `BIOCONDUCTOR_ONLINE_VERSION_DIAGNOS` to `FALSE` can speed package loading when internet access is slow or non-existent, but may result in out-of-date information regarding the current release and development versions of *Bioconductor*. In addition, offline organizations and its 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::repositories()` returns the *Bioconductor* and CRAN repositories used by `install()`.

`install.packages()` installs the packages themselves (used by `BiocManager::install` internally).

`update.packages()` updates all installed packages (used by `BiocManager::install` internally).

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

`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.

`containerRepository()` reports the location of the repository of binary packages for fast installation within containerized versions of Bioconductor, if available.

Usage

```
repositories(
  site_repository = character(),
  version = BiocManager::version(),
  ...,
  type = "both"
)

containerRepository(version = BiocManager::version(), type = "binary")
```

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.
`...`	Additional parameters passed to lower level functions, not used.
`type`	(Optional) `character(1)` indicating the type of package repository to retrieve (default: "both"). Setting `type` to "source" will disable any Bioconductor binary packages specifically built for the containers.

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.

To install binary packages on containerized versions of Bioconductor, a default binary package location URL is set as a package constant, see `BiocManager:::BINARY_BASE_URL`. Binary package installations are enabled by default for Bioconductor Docker containers. Anyone wishing to opt out of the binary package installation can set either the variable or the option, `BIOCONDUCTOR_USE_CONTAINER_REPOSITORY`, to `FALSE`. Note that the availability of Bioconductor package binaries is experimental and binary installations are intended to be used with `bioconductor/bioconductor_docker` images where such installations correspond to specific versions of Linux / Ubuntu.

If alternative default repositories are known to provide appropriate versions of CRAN or *Bioconductor* packages, the message may be silenced by setting either the option or the variable to `FALSE`, i.e., `options(BiocManager.check_repositories = FALSE)` or `BIOCMANAGER_CHECK_REPOSITORIES=FALSE`. Alternative default repositories are not guaranteed to work without issues related to incompatible package installations and are used at the user's own risk.

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()`. To set a more permenanent `site_repository`, one can use either the `BIOCMANAGER_SITE_REPOSITORY` environment variable or the `options(BiocManager.site_repository = ...)` option.

For greater flexiblity 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.

The unexported URL to the base repository is available with `BiocManager:::BINARY_BASE_URL`.

`BIOCONDUCTOR_USE_CONTAINER_REPOSITORY` is an environment variable or global `options()` which, when set to `FALSE`, avoids the fast installation of binary packages within containerized versions of Bioconductor.

Value

`repositories()`: named `character()` of repositories.

`containerRepository()`: character(1) location of binary repository, if available, or character(0) if not.

See Also

`BiocManager::install()` Installs or updates Bioconductor, CRAN, and GitHub packages.

`chooseBioCmirror()` choose an alternative Bioconductor mirror; not usually necessary.

`chooseCRANmirror()` choose an alternative CRAN mirror; not usually necessary.

`setRepositories()` Select additional repositories for searching.

Examples

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

## End(Not run)

containerRepository() # character(0) if not within a Bioconductor container
```

`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

`pkgs`	A character() vector of package names for checking, or a matrix as returned by `installed.packages()`‘.
`lib.loc`	A character() vector of library location(s) of packages to be validated; see `installed.packages()`.
`priority`	character(1) Check validity of all, "base", or "recommended" packages; see `installed.packages()`.
`type`	character(1) The type of available package (e.g., binary, source) to check validity against; see `available.packages()`.
`filters`	character(1) Filter available packages to check validity against; see `available.packages()`.
`...`	Additional arguments, passed to `BiocManager::install()` when `fix=TRUE`.

`checkBuilt`	`logical(1)`. If `TRUE` a package built under an earlier major.minor version of R (e.g., 3.4) is considered to be old.
`site_repository`	`character(1)`. See `?install`.
`x`	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::install()` to update installed packages.

Examples

```
if (interactive()) {
  BiocManager::valid()
}
```

`version` *Version of Bioconductor currently in use.*

Description

`version()` reports the version of *Bioconductor* appropropriate 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 `unknown_version` 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.

The environment variable `R_BIOC_VERSION` can be used to specify a version that is not consistent with *Bioconductor* release versioning. Use of this variable is strongly discouraged.

Value

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

Examples

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

Index