Package ‘BiocCheck’

October 16, 2024

Version 1.40.0

Title Bioconductor-specific package checks

Description BiocCheck guides maintainers through Bioconductor best practicies. It runs Bioconductor-specific package checks by searching through package code, examples, and vignettes. Maintainers are required to address all errors, warnings, and most notes produced.

Depends R (>= 4.3.0)

License Artistic-2.0

VignetteBuilder knitr

Imports BiocBaseUtils, BiocFileCache, BiocManager, biocViews (>= 1.33.7), callr, codetools, graph, httr2, knitr, methods, rvest, stringdist, tools, utils

Suggests RUnit, BiocGenerics, Biobase, jsonlite, rmarkdown, downloader, devtools (>= 1.4.1), usethis, BiocStyle, GenomicRanges, gert

Date 2024-04-25

biocViews Infrastructure

URL `https://github.com/Bioconductor/BiocCheck`

BugReports `https://github.com/Bioconductor/BiocCheck/issues`

RoxygenNote 7.3.1

Encoding UTF-8

git_url https://git.bioconductor.org/packages/BiocCheck

git_branch RELEASE_3_19

git_last_commit ca6d2f9

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-10-16

Author Bioconductor Package Maintainer [aut],
Lori Shepherd [aut],
Daniel von Twisk [ctb],
Kevin Rue [ctb],
Marcel Ramos [aut, cre] (<https://orcid.org/0000-0002-3242-0582>),
Leonardo Collado-Torres [ctb],
Federico Marini [ctb]

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

Contents

`BiocCheck` *Check a package's adherence with the Bioconductor Package Guidelines*

Description

Analyzes an R package for adherence with Bioconductor package guidelines and best practices. The check outputs are categorized into ERROR, WARNING, and NOTE. See the vignette for more details. `BiocCheck` is complementary to `R CMD check`, which should always be run first.

Usage

```
BiocCheck(
  package = getwd(),
  checkDir = dirname(package),
  debug = FALSE,
  callr = FALSE,
  ...
)
```

Arguments

`package`	The path to an R package directory or tarball (`.tar.gz`). The `BiocCheck` function is intended to be run from the package directory; therefore, the current working directory (given by `getwd()`) is the default.

`checkDir`	The directory where the `BiocCheck` output directory will be stored. By default, it will be placed in the same directory as the package directory i.e., `dirname(pkg_dir)`.
`debug`	Whether to append the names of functions that correspond to each condition raised by `BiocCheck` in the written log (i.e., in the `'<package_name>.BiocCheck'` folder). This option is only relevant to developers and contributors to `BiocCheck`.
`callr`	logical(1) Whether to use the `callr` package to run `BiocCheck` in an isolated R session to prevent namespace collisions.
`...`	See the details section for available options. When running `BiocCheck`, options can be specified as:

```
BiocCheck(package, `no-check-vignettes`=TRUE)
```

Details

`BiocCheck()` reviews R packages for adherence with Bioconductor package guidelines and best practices. See `https://contributions.bioconductor.org` for the latest guidance for writing Bioconductor software. Some rationale behind these best practices can be seen in the vignette and pages in the `references` section. The vignette also provides detailed explanations of all the checks performed by `BiocCheck`.

`BiocCheck` is called within R with

```
BiocCheck(<package>)
```

where `package` points to the source directory or the `.tar.gz` tarball that was created using `R CMD build`.

Note that `BiocCheck` is complementary to `R CMD check`. `R CMD check` should always be run first for best results.

Value

`BiocCheck()` is chiefly called for the side effect of the check reporting. The function also creates a `<package_name>.BiocCheck` folder and returns a `BiocCheck` reference class with three main list elements:

`error`	Items to address before the package can be accepted
`warning`	Strongly suggested items that may require attention
`note`	Items to consider, though not required, before acceptance

dot-options

new-package enable checks specific to new packages

no-check-dependencies disable check for bad dependencies

no-check-deprecated disable check for usage of deprecated packages

no-check-remotes disable check for usage of remote packages other than those hosted on CRAN or Bioconductor

no-check-version-num disable check for valid version number

no-check-R-ver disable check for valid R version

no-check-pkg-size disable check for package tarball size

no-check-file-size disable check for individual file size

no-check-bioc-views disable biocViews-specific checks (for non-BioC packages)

no-check-bbs disable BBS-specific checks (for non-BioC packages). Valid DESCRIPTION

no-check-description disable DESCRIPTION file checks

no-check-vignettes disable vignette checks

no-check-library-calls disable check usage of functions that install or update packages

no-check-install-self disable check for require or library of itself

no-check-coding-practices disable check for some common best coding practices

no-check-function-len disable check for function length

no-check-man-doc disable checks for man page documentation

no-check-news disable checks for NEWS file

no-check-unit-tests disable checks for unit tests

no-check-skip-bioc-tests disable check for tests that skip when on bioc

no-check-formatting disable checks for file formatting

no-check-CRAN disable check for if package exists in CRAN

no-check-bioc-help disable check for registration on Bioconductor mailing list and support site

build-output-file file containing R CMD build output, for additional analysis

quit-with-status enable exit code option when performing check

Author(s)

Dan Tenenbaum, Lori Shepherd, and Marcel Ramos

References

`https://contributions.bioconductor.org`

See Also

BiocCheck-class, Message-class

Examples

```
packageDir <- system.file("testpackages", "testpkg0", package="BiocCheck")
BiocCheck(packageDir, `quit-with-status`=FALSE)
```

`BiocCheck-class` *A class for composing BiocCheck reports.*

Description

The `BiocCheck` class provides a framework for reporting checks based on Bioconductor guidelines. The class has several methods for working with the provided checks that handle and display messages and the display of the metadata. These methods also record the output of the `BiocCheck()` report in both plain text and JSON formats.

Note that currently, multiple `BiocCheck` runs will interfere with each other given that they are implemented via a reference class semantic. When running multiple checks in the same session, you can separate these instances by running them in separate processes (e.g., via `BiocParallel`).

Details

The metadata includes a number of standard fields to allow easier troubleshooting and display of potentially relevant information. Currently, the fields included are

BiocCheckVersion The version of the BiocCheck package

BiocVersion The version of Bioconductor

Package The name of the package in check

PackageVersion The version of the package in check

sourceDir The directory of the package source or tarball in check

installDir The directory where the package is installed for testing, a temporary location by default

BiocCheckDir The directory where the `<package>.BiocCheck` folder is saved. Usually the same folder as the package in check

platform The platform/OS where the check is taking place

isTarBall Whether the package in check is a source directory or a tarball

Value

A `BiocCheck` instance

Fields

`log list()` A running list of all conditions raised (i.e., notes, warnings, errors)

`check character(1)` The title of the last check used for logging purposes.

`error,` warning, note `list()` Finer extraction of each condition type

`metadata list()` A list of additional information relevant to the package and its state. See details.

`verbose logical(1)` Whether to show the full information pertaining to the checks. A `FALSE` value will only show the condition messages and not any relevant files or additional information. The defaults are `FALSE` and `TRUE` for `BiocCheck` and `BiocCheckGitClone`, respectively.

See Also

Message-class

Examples

```
bc <- BiocCheck:::.BiocCheck

bc$verbose
```

`BiocCheck-methods` *A list of methods for the BiocCheck reference class*

Description

A list of methods for the BiocCheck reference class

Arguments

`...`	`character()` A vector that makes up the `BiocCheck` exception message (e.g., 'Vignette must be built by R CMD build'). The character vector is handled with `paste0` and made into a list and appended with `help_text` and `messages`.
`help_text`	`character(1)` Additional text prompting a list of files (e.g,. "Found in files:")
`condition`	`character(1)` One of the three conditions handled: `error`, `warning`, or `note`
`messages`	`character()` Often a vector of file names where the check was triggered.
`verbose`	`logical(1)` Whether or not to output both the `help_text` and `messages` as part of the report
`debug`	`logical(1)` Whether to append the name of the originating check name into for trace-ability
`checkName`	`character(1)` The title of the current group of checks. It can be set with `handleCheck`, e.g., `handleCheck("Checking for version number mismatch...")`. Internally, it ' is saved with `setCheck` and obtained with `getLastCheck`.
`isOnBBS`	`logical(1)` Indicates whether the checks are being run on the Bioconductor Build System (BBS). This is helpful for avoiding the creation of folders in the BBS.
`file`	`character(1)` A path to a JSON file for writing or reading as created by `toJSON` and `fromJSON` `BiocCheck` methods.

Value

An internal `BiocCheck` R5 Reference Class used to document conditions such as errors, warnings, and notes

methods

- **add** Include a condition to the `BiocCheck` report
- **getLastCheck** Obtain the name of the last check run
- **setCheck** Create a new element in the internal list for a check
- **get** Extract the list of conditions raised by `BiocCheck`
- **getNum** Tally the number of condition provided by the input
- **zero** Reset the internal log of the condition provided
- **getBiocCheckDir** Report and create the `<package>.BiocCheck` directory as obtained from the metadata
- **composeReport** Simplify the list structure from the `log` and provide a character vector of conditions raised
- **report** Write the `00BiocCheck.log` report into the `BiocCheck` folder
- **toJSON** Write a JSON file to the location indicated with the conditions raised
- **fromJSON** Read a JSON file from the location indicated with the output of previous conditions raised in the check
- **show** Display the information in the class. Currently empty.
- **show_meta** Display the metadata information stored in the `metadata` field

---

`BiocCheckGitClone` *Checks specific to a Git clone of a package repository*

Description

Analyzes an R package for adherence with Bioconductor package guidelines and best practices. The check outputs are categorized into ERROR, WARNING, and NOTE. This function is typically used in the Bioconductor Build System (BBS) and not intended for general use.

Usage

```
BiocCheckGitClone(package = ".", ...)
```

Arguments

`package`	A directory containing an R source package. Not a package tar ball.
`...`	Currently, only `quit-with-status` is available. See `BiocCheck`

Details

`BiocCheckGitClone()` reviews R packages for adherence with Bioconductor package guidelines and best practices. See `https://contributions.bioconductor.org` for the latest guidance for writing Bioconductor software. This function should only be run on a source directory and not on a tarball.

`BiocCheckGitClone` is called within R with, as

`BiocCheckGitClone(<package>)`

where package is the source directory containing the `R` package.

Value

`BiocCheckGitClone()` is chiefly called for the side effect of the check reporting. The function returns a `BiocCheck` reference class with three main list elements:

`error`	Items to address before the package can be accepted
`warning`	Strongly suggested items that may require attention
`note`	Items to consider, though not required, before acceptance

Author(s)

Lori Shepherd

References

`https://contributions.bioconductor.org`

See Also

BiocCheck-class

Examples

```
packageDir <- system.file("testpackages", "testpkg0", package="BiocCheck")
BiocCheckGitClone(packageDir, `quit-with-status`=FALSE)
```

`Context` *Report context of events to user with a data.frame of events and locations*

Description

Report context of events to user with a data.frame of events and locations

Usage

```
Context(file = "", lines = character(), idx = logical(), offset = 0L)
```

Arguments

`file`	character(1) full path (including package name) of file being summarized.
`lines`	character() vector of text lines in file
`idx`	logical() same length as `lines` indicating lines in which event occurs
`offset`	‘integer(1)‘ The number of lines to add to the ’Line’ column calculation. It is mainly used to account for the number of lines that the YAML header occupies.

Value

Context: a data.frame() with columns File, Line, and Context

`Message-class` *A lower level Message helper class for BiocCheck*

Description

A lower level Message helper class for BiocCheck

Value

A `Message` class instance

Fields

`msg` `list()` A list of character messages usually grown with append with conditions raised by a check

`condition` character(1) One of the three conditions handled: `error`, `warning`, or `note`

See Also

BiocCheck-class

`Message-methods` *A list of methods for the Message reference class*

Description

A list of methods for the Message reference class

Arguments

`condition`	character(1) One of the three conditions handled: `error`, `warning`, or `note`
`verbose`	logical(1) Whether to output the full text in the check or only the check name itself in the report
`...`	`list()` A nested list with the check name as the top level layer. Second level lists include any `help_text` and `messages` that are part of the check.

Value

An internal `R5` Reference Class to handle messages and their conditions, e.g., for errors, warnings, or notes.

Index