

# Package ‘BiocCheck’

May 15, 2025

**Title** Bioconductor-specific package checks

**Version** 1.45.6

**Date** 2025-05-02

**Description** BiocCheck guides maintainers through Bioconductor best practices. 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.

**License** Artistic-2.0

**URL** <https://github.com/Bioconductor/BiocCheck>

**BugReports** <https://github.com/Bioconductor/BiocCheck/issues>

**Depends** R (>= 4.4.0)

**Imports** BiocBaseUtils, BiocFileCache, BiocManager, biocViews, callr, cli, codetools, graph, httr2, knitr, methods, rvest, stringdist, tools, utils

**Suggests** BiocStyle, devtools, gert, jsonlite, rmarkdown, tinytest, usethis

**VignetteBuilder** knitr

**biocViews** Infrastructure

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**git\_url** <https://git.bioconductor.org/packages/BiocCheck>

**git\_branch** devel

**git\_last\_commit** d1618aa

**git\_last\_commit\_date** 2025-05-02

**Repository** Bioconductor 3.22

**Date/Publication** 2025-05-15

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

Leonardo Collado-Torres [ctb],  
 Federico Marini [ctb]

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

## Contents

BiocCheck . . . . .	2
BiocCheck-class . . . . .	4
BiocCheckGitClone . . . . .	6
BiocPackage-class . . . . .	7
Context . . . . .	9
Message-class . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

BiocCheck	<i>Check a package's adherence with the Bioconductor Package Guidelines</i>
-----------	---

---

## 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

*An internal 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`).

### Arguments

<code>...</code>	<code>character()</code> A vector that makes up the <code>BiocCheck</code> exception message (e.g., 'Vignette must be built by R CMD build'). The character vector is handled with <code>paste0</code> and made into a list and appended with <code>help_text</code> and messages.
<code>help_text</code>	<code>character(1)</code> Additional text prompting a list of files (e.g., "Found in files:")
<code>condition</code>	<code>character(1)</code> One of the three conditions handled: error, warning, or note
<code>messages</code>	<code>character()</code> Often a vector of file names where the check was triggered.
<code>debug</code>	<code>logical(1)</code> 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 <code>handleCheck</code> , e.g., <code>handleCheck("Checking for version number mismatch...")</code> . Internally, it ' is saved with <code>setCheck</code> and obtained with <code>getLastCheck</code> .
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 <code>toJSON</code> and <code>fromJSON</code> <code>BiocCheck</code> methods.

### 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

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

### 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.

### methods

- `add`: Include a condition to the `BiocCheck` report
- `addMetadata`: Add metadata to the `BiocCheck` object from a `BiocPackage` object
- `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

### See Also

[Message-class](#), [BiocPackage-class](#)

### Examples

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

---

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

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

### 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)
```

---

BiocPackage-class      *A class for representing files in a Bioconductor package*

---

**Description**

The BiocPackage class is used to represent a Bioconductor package. It is used by BiocCheck to store information about the package being checked. The class has several methods to identify the type of package, check for common issues, and store metadata about the package.

**Usage**

```
.BiocPackage
```

**Format**

An object of class BiocPackage of length 1.

**Value**

An object of class BiocPackage

**Fields**

isValid logical indicating whether the package's DESCRIPTION file was able to be read without any errors  
 isTar logical indicating whether the package is a tarball  
 isSourceDir logical indicating whether the package being checked is from a source directory  
 isInfrastructure logical indicating whether the package is an Bioconductor infrastructure package based on the biocViews field  
 usesRoxygen logical indicating whether the package uses roxygen2 documentation  
 usesRdpack logical indicating whether the package uses Rdpack package  
 DESCRIPTION matrix containing the DCF contents of the DESCRIPTION file  
 dependencies character vector of package dependencies  
 readError character error message if the DESCRIPTION file could not be read  
 packageVersion character version of the package  
 packageType character indicating the type of package based on the biocViews field; can be NA\_character\_ there are invalid biocViews terms  
 sourceDir character path to the source directory  
 vignettesDir character path to the vignettes directory  
 RSources character vector of R source files  
 VigSources character vector of vignette source files  
 manSources character vector of Rd source files  
 BiocCheckDir character path to the directory where the package BiocCheck logs are written  
 packageName character name of the package  
 tarFilename character filename of the tarball  
 metadata list containing metadata about the package

**methods**

- initialize: Initialize a BiocPackage object
- getPackageDir: Get the package directory
- getRSources: Get the R source files
- getVigSources: Get the vignette source files
- getManSources: Get the Rd source files
- getBiocCheckDir: Get the directory where the BiocCheck logs are written
- getBiocViews: Get the biocViews field from the DESCRIPTION file
- getPackageType: Get the package type based on the biocViews field
- readDESCRIPTION: Read the DESCRIPTION file
- getVigBuilder: Get the vignette builder
- getAllDependencies: Get all dependencies from the DESCRIPTION file
- findInfrastructure: Is the package an infrastructure package?
- findRoxygen: Does the package use roxygen2?
- getPackageVersion: Get the package version
- untarTarball: Untar the source tarball



**See Also**

[BiocCheck-class](#), [Message-class](#)

**Examples**

```
# Create a BiocPackage object
packageDirectory <- "path/to/package"
if (dir.exists(packageDirectory))
  .bioctest <- .BiocPackage$initialize(packageDirectory)

.bioctest <- BiocCheck::.BiocPackage

.bioctest$DESCRIPTION
```

---

Context	<i>Report context of events to user with a data.frame of events and locations</i>
---------	---

---

**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	<i>A lower level Message helper class for BiocCheck</i>
---------------	---

---

**Description**

A lower level Message helper class for BiocCheck

**Arguments**

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

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

**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

**methods**

- setMessage: Set the message and condition (error, warning, or note) for the check
- setCondition: Change the condition for the check
- getCondition: Get the condition from the Message class

**See Also**

[BiocCheck-class](#) [BiocPackage-class](#)

# Index

## \* **internal**

BiocPackage-class, [7](#)

Context, [9](#)

Message-class, [10](#)

.BiocPackage (BiocPackage-class), [7](#)

.MessageCondition (Message-class), [10](#)

BiocCheck, [2](#)

BiocCheck-class, [4](#), [4](#), [7](#), [9](#), [10](#)

BiocCheckGitClone, [6](#)

BiocPackage (BiocPackage-class), [7](#)

BiocPackage-class, [6](#), [7](#), [10](#)

Context, [9](#)

getCondition, Message-method  
(Message-class), [10](#)

Message-class, [4](#), [6](#), [9](#), [10](#)

setCondition, Message-method  
(Message-class), [10](#)

setMessage, Message-method  
(Message-class), [10](#)