

Package ‘BiocPkgDash’

April 10, 2026

Type Package

Title An interactive Shiny dashboard for Bioconductor package maintainers

Version 0.99.49

Description This package provides an interactive Shiny dashboard for Bioconductor package maintainers. It visualizes various package statuses, metadata, and development metrics, offering insights into package health and activity. This tool aims to support maintainers of multiple packages by filtering packages via maintainer email.

Depends R (>= 4.6.0), shiny

Imports BiocBaseUtils, BiocPkgTools (>= 1.27.6), BiocManager, bsicons, bslib, dplyr, DT, ggplot2, gh, htmlwidgets, plotly, rmarkdown, shinybiocloader, shinyjs, tibble, tidyr, utils, whisker, yaml

Suggests BiocStyle, knitr, sessioninfo, tinytest

biocViews Software, Infrastructure, Visualization, GUI

VignetteBuilder knitr

License Artistic-2.0

Encoding UTF-8

URL <https://github.com/Bioconductor/BiocPkgDash>

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

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

Date 2026-03-23

git_url <https://git.bioconductor.org/packages/BiocPkgDash>

git_branch devel

git_last_commit 3650cbd

git_last_commit_date 2026-03-23

Repository Bioconductor 3.23

Date/Publication 2026-04-10

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

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

Contents

BiocPkgDash	2
pkgStatusPlot	3
pkgStatusTable	4
Index	5

BiocPkgDash	<i>The Bioconductor Package Dashboard</i>
-------------	---

Description

A dashboard for Bioconductor package maintainers to monitor the status of their packages. The key input to the dashboard is the email the maintainer uses in their package's DESCRIPTION file. The dashboard displays the badge statuses of the package in both the release or devel branches of Bioconductor. The status is determined by the results of the Bioconductor nightly builds. The dashboard also provides a visualization of the status of the package checks on either the release or devel branches of Bioconductor.

Usage

```
BiocPkgDash(email = NULL, ...)
```

Arguments

email	character(1) vector providing the email of the maintainer. By default, the package will look for the email in the URL query.
...	Additional parameters to pass to the <code>shiny::shinyApp()</code> function.

Value

called for the side effect of initializing a shiny app

Examples

```
BiocPkgDash()
```

`pkgStatusPlot`*A Summary Plot for Package Statuses*

Description

This function generates a stacked bar plot of package statuses for a given Bioconductor version and email combination. It is mainly used for the Bioconductor Package Dashboard.

Usage

```
pkgStatusPlot(  
  data = NULL,  
  status = c("OK", "WARNINGS", "ERROR", "TIMEOUT", "skipped"),  
  stage = c("install", "buildsrc", "checksrc", "buildbin")  
)
```

Arguments

<code>data</code>	<code>tibble()</code> / <code>data.frame()</code> A data frame of maintained packages. This is used internally to avoid repeated calls to the <code>BiocPkgTools::biocMaintained()</code> function.
<code>status</code>	<code>character()</code> A vector of INSTALL, build and check statuses to include in the plot. These values are obtained from the result column in <code>BiocPkgTools::biocBuildReport()</code> . The default is all: <code>c("OK", "WARNINGS", "ERROR", "TIMEOUT", "skipped")</code> .
<code>stage</code>	<code>character()</code> A vector of the Bioconductor Build System (BBS) stages to include in the plot. These values are obtained from the stage <code>BiocPkgTools::biocBuildReport()</code> . The default is all stages: <code>c("install", "buildsrc", "checksrc", "buildbin")</code> .

Details

Note that binary build stages for the Linux builders are not included in the plot. This is because the binaries are built on GitHub Actions and their result are not included in the Bioconductor Build System (BBS) database.

Value

An interactive ggplotly object.

Examples

```
data <- BiocPkgTools::biocMaintained(  
  "maintainer@bioconductor.org", pkgType = "software"  
)  
pkgStatusPlot(data)
```

pkgStatusTable	<i>Build a table of package build statuses</i>
----------------	--

Description

This function builds a table of package build statuses for a given Bioconductor version and email combination. It is mainly used for the Bioconductor Package Dashboard.

Usage

```
pkgStatusTable(  
  data = NULL,  
  status = c("OK", "WARNINGS", "ERROR", "TIMEOUT", "skipped"),  
  stage = c("install", "buildsrc", "checksrc", "buildbin")  
)
```

Arguments

data	tibble() / data.frame() A table of maintained packages. This is used internally to avoid repeated calls to the <code>BiocPkgTools::biocMaintained()</code> function.
status	character() The status of the builders to include in the table. These values are obtained from the <code>result</code> column in <code>BiocPkgTools::biocBuildReport()</code> . The default is all: <code>c("OK", "WARNINGS", "ERROR", "TIMEOUT", "skipped")</code> .
stage	character() A vector of the Bioconductor Build System (BBS) stages to include in the plot. These values are obtained from the <code>stage</code> column in <code>BiocPkgTools::biocBuildReport()</code> . The default is all stages: <code>c("install", "buildsrc", "checksrc", "buildbin")</code> .

Details

Note that binary build stages for the Linux builders are marked as skipped in the table. This is because the binaries are built on GitHub Actions and their result are not included in the Bioconductor Build System (BBS) database. Provide the data argument to avoid recomputing the list of maintained packages for a given email and Bioconductor version. Annotation packages are not included in the table because they are not built regularly by the BBS.

Value

A tibble() / data.frame() with the package build statuses for the given data input.

Examples

```
data <- BiocPkgTools::biocMaintained("maintainer@bioconductor.org")  
pkgStatusTable(data)
```

Index

BiocPkgDash, [2](#)

BiocPkgTools::biocBuildReport(), [3](#), [4](#)

BiocPkgTools::biocMaintained(), [3](#), [4](#)

pkgStatusPlot, [3](#)

pkgStatusTable, [4](#)

shiny::shinyApp(), [2](#)