Package 'BiocBook'

January 20, 2025

Title Write, containerize, publish and version Quarto books with Bioconductor

Description A BiocBook can be created by authors (e.g. R developers, but also scientists, teachers, communicators, ...) who wish to 1) write (compile a body of biological and/or bioinformatics knowledge), 2) containerize (provide Docker images to reproduce the examples illustrated in the compendium), 3) publish (deploy an online book to disseminate the compendium), and 4) version (automatically generate specific online book versions and Docker images for specific Bioconductor releases).

Version 1.5.0

Date 2023-08-03

URL https://bioconductor.org/packages/BiocBook

BugReports https://github.com/js2264/BiocBook/issues

Depends R (>= 4.3)

Imports BiocGenerics, available, cli, glue, gert, gh, gitcreds, httr, usethis, dplyr, purrr, tibble, methods, rprojroot, stringr, yaml, tools, utils, rlang, quarto, renv

Suggests BiocStyle, knitr, testthat (>= 3.0.0), rmarkdown

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

biocViews Infrastructure, ReportWriting, Software

VignetteBuilder knitr

License MIT + file LICENSE

Collate 'doc.R' 'imports.R' 'init.R' 'BiocBook.R' 'BiocBook-methods.R' 'check.R' 'editing.R' 'globals.R' 'pages.R' 'utils.R'

Config/testthat/edition 3

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

git_branch devel

git_last_commit 0148dc3

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2025-01-20

Author Jacques Serizay [aut, cre]

Maintainer Jacques Serizay <jacquesserizay@gmail.com>

Contents

`BiocBook` *Handling BiocBook directories*

Description

`BiocBooks` are local R packages containing an extra `pages` folder to write up online book chapters.

1. A new `BiocBook` should be created using `init(new_package = "...")` (or alternatively, locally with `quick_init(new_package = "...", user = "...")`).
2. A newly created `BiocBook` can be accessed to in R using `biocbook <- BiocBook(path = "...")`.
3. To edit an existing `BiocBook` object, several helper functions exist:

- `add_preamble(biocbook)`
- `add_chapter(biocbook, title = "...")`
- `edit_page(biocbook, page = "...")`

Read the different sections down below for more details.

Usage

```
init(new_package, push = NA, skip_availability = FALSE, .local = FALSE)

quick_init(new_package, user)

BiocBook(path)

releases(object)

chapters(object)

## S4 method for signature 'BiocBook'
```

```
path(object)

## S4 method for signature 'BiocBook'
releases(object)

## S4 method for signature 'BiocBook'
chapters(object)

## S4 method for signature 'BiocBook'
show(object)
```

Arguments

new_package	Name to use when initiating a new BiocBook. This name should be compatible with package naming conventions from R and Bioconductor (i.e. no _ or -, no name starting with a number).
push	Optional. Logical, whether to automatically push commits to remote Github origin. If NA, a prompt will ask whether to push commits or not (default: NA).
skip_availability	Optional. Whether to skip package name availability (default: FALSE).
.local	Should only be used for examples/tests. Whether to create a matching Github repository or stay local (default: FALSE).
user	Ideally, the Github ID of the main author/organization.
path	Path of an existing BiocBook.
object	A BiocBook object, created by BiocBook or init().

Value

- init("newBook") creates a local directory, synchronizes it with the registered GitHub user, and invisibly returns a BiocBook object.
- BiocBook("newBook") returns a BiocBook object.
- path(bb), releases(bb) and chapters(bb) return the corresponding information related to the bb BiocBook.

The BiocBook class

A BiocBook object acts as a pointer to a local package directory, with book chapters contained in a pages/ folder as .qmd files.

This package directory requires a specific architecture, which is best set up using the init() function.

When created, 3 slots are defined:

- title: The title contained in /inst/assets/_book.yml
- local_path: The absolute path to the book package directory
- remote_repository: If the book is synced with Github, this will indicate the remote

Creating a BiocBook

A new BiocBook should be created using the `init(new_package = "...")` function. This function performs the following operations:

1. It checks that the provided package name is available;
2. It logs in the GitHub user accounts;
3. It creates a new **local** repository using the BiocBook template from `js2264/BiocBook`;
4. It pushes the local repository to a **remote** Github repository;
5. It creates an empty `gh-pages` and sets it up to serve rendered books;
6. It edits several placeholders from the template and commits the changes.

The `init(new_package = "...")` function returns a BiocBook object.

Quickly create a local BiocBook

Alternatively, a **local** BiocBook can be quickly created using the `quick_init(new_package = "...", user = "...")` function.

This function only creates a new **local** repository, using the BiocBook template from `js2264/BiocBook`.

It does **NOT**:

- Check that the provided package name is available;
- Set up/push the local repository to a **remote** Github repository;
- Set up a `gh-pages` to serve rendered books;

This implies that functions committing/pushing (`publish()`) or checking remote status (`status()`) do not work properly with a BiocBook initiated with `quick_init()`, unless a remote is manually set up.

To enable Github support for a local BiocBook, one has to manually initiate a `git` repository and add a remote as follows:

```
git init
git symbolic-ref HEAD refs/heads/devel
git add .
git commit -m 'first commit'
git remote add origin git@github.com:<user>/<biocbook>.git
git push --set-upstream origin devel
```

Editing an existing BiocBook

BiocBook objects can be modified using the following helper functions:

- `add_preamble(biocbook)` to start writing a preamble;
- `add_chapter(biocbook, title = "...")` to start writing a new chapter;
- `edit_page(biocbook, page = "...")` to edit an existing chapter.

Publishing an existing BiocBook

Important: remember to add any dependency used in your chapters to the DESCRIPTION before publishing your book. Dependencies across chapters can be found with:

```
check_deps(biocbook)
```

Note that this will not always work 100%, always use good coding practices and add your dependencies to DESCRIPTION while writing new chapters.

To locally preview the book, one can use the following command:

```
preview(biocbook)
```

To publish changes, as long as the local BiocBook has been initiated with `init()`, the writer simply has to commit changes and push them to the `origin` remote. In R, this can be done as follows:

```
publish(biocbook)
```

The different available versions published in the `origin` `gh-pages` branch can be listed with

```
status(biocbook)
```

Examples

```
## In practice, you should not use `.local` argument.
unique_id <- sample(c(LETTERS, 0:9), 8) |> paste(collapse = '')
bookname <- paste(Sys.info()[['sysname']], unique_id, sep = '.')
init(bookname, .local = TRUE)
bb <- BiocBook(bookname)
chapters(bb)
releases(bb)
unlink(bookname, recursive = TRUE)
```

BiocBook-editing *Editing BiocBook accessory files*

Description

Editing functions for BiocBooks. See BiocBook help sections for extended description.

Usage

```
check_deps(book)

edit_yml(book, yml = c("_book", "_website", "_knitr", "_format"), open = TRUE)

edit_bib(book, open = TRUE)

edit_requirements_yml(book, open = TRUE)

edit_css(book, open = TRUE)

preview(book, browse = FALSE, watch = FALSE)
```

```
publish(book, message = "Publishing")

status(book)

add_preamble(book, open = TRUE)

add_chapter(book, title, file = NA, position = NULL, open = TRUE)

edit_page(book, file, open = TRUE)
```

Arguments

book	A BiocBook object, opened with BiocBook or created by init().
yml	Which .yml should be opened?
open	Optional. Whether to open the file for interactive editing (default: TRUE)
browse	Optional. Passed to quarto_preview() (default: FALSE).
watch	Optional. Passed to quarto_preview() (default: FALSE).
message	Optional. Message used when committing with publish().
title	A character string for a title for the new chatper. If file is not explicitely provided, the title should only contain alphanumeric characters and spaces
file	Optional. A character string for the name of the .qmd file to write the new chapter. The extension .qmd has to be provided. If not provided, the file name is deduced from the title argument.
position	Optional. A position to insert the chapter. For example, if position = 2, the new chapter will be inserted after the first existing chapter (i.e. the Welcome page)

Value

- add_*, edit_*: A BiocBook object (invisible).
- publish: TRUE (invisible) if pushing to Github was successful;
- preview: Local URL to browse dynamically rendered book;
- status: A tibble of the existing versions found on the Github repository (branch gh-pages) and of the existing Dockerfiles.

add_* functions

add_chapter() and add_preamble are convenient functions to add pages to a BiocBook.

edit_* functions

Several accessory files can be manually edited:

- edit_page(): manually edit any page listed in chapters(book)
- edit_bib(): manually edit /inst/assets/bibliography.bib
- edit_yml(): manually edit the different yml in /inst/assets/
- edit_requirements_yml(): manually edit /inst/requirements.yml

Maintenance functions

Extra functions are provided to faciliate the maintenance of BiocBooks.

- check_deps(): is used to find dependencies from chapter pages that are not listed in DESCRIPTION
- preview(): is used to dynamically render the book locally
- publish(): is used to commit and push to remote Github branch
- status(): is used to list the book versions already deployed on the Github repository (branch gh-pages) and of the existing Dockerfiles

See Also

BiocBook

Examples

```
## In practice, you should not use `.local` argument.
unique_id <- sample(c(LETTERS, 0:9), 8) |> paste(collapse = '')
bookname <- paste(Sys.info()[['sysname']], unique_id, sep = '.')
bb <- init(bookname, .local = TRUE)
add_preamble(bb, open = FALSE)
add_chapter(bb, title = "Chapitre Un", open = FALSE)
unlink(bookname, recursive = TRUE)
```

Index