Write Quarto books with Bioconductor

Table of contents

What are BiocBooks?					
W	hat is	s this {BiocBook} package?	6		
Main features of BiocBooks Fully compatible with the Bioconductor Build System					
Αc	know	vledgments	9		
Se	ssion	info	10		
Re	ferer	ices	13		
Pr	eaml	ole	14		
1	Bio(CBook versioning system Continuous Integration and Continuous Delivery	15 15 15		
		1.1.2 Package submission to Bioconductor	16 18		
	1.2	1.1.4 Updates	19 20 20		
	1.3 1.4	1.2.2 Website versioning	21 21 21		
2	Writing a {BiocBook} package				
	2.1	Register a Github account in R	23 23 23 24		

	2.2	BiocBook workflow	24						
		2.2.1 Initiate a {BiocBook} package							
		2.2.2 Edit new BiocBook chapters	29						
		2.2.3 Edit assets	29						
		2.2.4 Previewing and publishing changes	30						
	2.3	Writing features	30						
		2.3.1 Executing code	30						
		2.3.2 Creating data object	31						
		2.3.3 Adding references	31						
3	Containerizing and Publishing against specific Bioconductor release versions 32								
	3.1		32						
	3.2	For a $\{BiocBook\}$ package accepted in Bioconductor > 6 months ago	32						
4	{Bio	ocBook}-based books vs. {rebook}-based books	34						
	4.1	Differences with {rebook}-based books	34						
	4.2	BiocBook features missing from rebook							
	4.3	rebook features missing from BiocBook	35						

What are BiocBooks?

Package: BiocBookDemo Authors: Jacques Serizay [aut, cre] Compiled: 2024-08-15 Package version: 1.3.0 R version: R version 4.4.1 (2024-06-14) BioC version: 3.20 License: MIT + file LICENSE

BiocBooks are package-based, versioned online books with a supporting Docker image for each book version.

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;
- 4. Version: automatically generate specific online book versions and Docker images for specific Bioconductor releases.



A {BiocBook}-based package hosted on **GitHub** with a branch named RELEASE_X_Y provides:

- A **Docker image**: hosted on ghcr.io;
- An **online book** (a.k.a website): hosted on the GitHub repository **gh-pages** branch;

Both are built against the specific Bioconductor release X.Y.

A {BiocBook}-based package submitted to **Bioconductor** also lead to the online book being independently built by the **Bioconductor Build System (BBS)** and deployed to https://bioconductor.org/books/

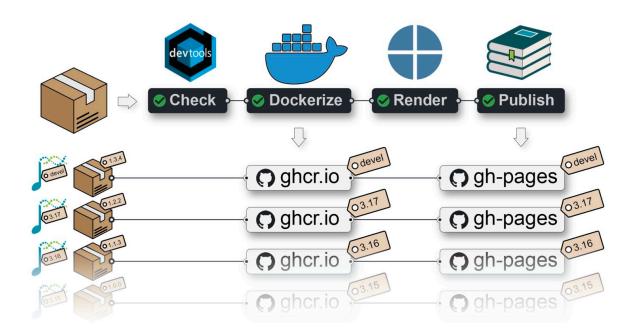
Sioconductor.org/books/

Sioconductor.org/books/

Sioconductor.org/books/

Sioconductor.org/books/

Sioconductor.org/books/



What is this {BiocBook} package?

The {BiocBook} package offers a streamlined approach to creating BiocBooks, with several important benefits:

- The author creates a {BiocBook}-based package without leaving R;
- The author writes book chapters in pages/*.qmd files using enhanced markdown;
- The author can submit its {BiocBook}-based package to Bioconductor.

The containerization and publishing of the new {BiocBook}-based package is automated:

- A Github Action workflow **generates different Docker images** for different Bioconductor releases, with the packages used in the book pre-installed;
- A Github Action workflow publishes different book versions for different Bioconductor releases.

Main features of BiocBooks

Fully compatible with the *Bioconductor Build System*

When a {BiocBook}-based package is accepted into Bioconductor, it is automatically integrated into the Bionconductor Build System (BBS).

This means that it is getting built using R CMD build --keep-empty-dirs --no-resave-data This triggers the rendering of the book contained in /inst/. built by the BBS are then automatically deployed and are eventually available at https://bioconductor.org/books/<bioc_version>/<pkg>/.

Automated versioning of Docker images

A separate Docker image is built for each branch (named devel or RELEASE_X_Y) of a {BiocBook}-based **Github repository**.

Each Docker image provides pre-installed R packages:

- Bioconductor release X.Y;
- Specific book dependencies from Bioconductor release X.Y (listed in DESCRIPTION);
- The book package itself

The Docker images also include a micromamba-based environment, named BiocBook, in which all the softwares listed in requirements.yml are installed.

For example, Docker images built from the {BiocBookDemo} package repository are available here:

ghcr.io/js2264/biocbookdemo



Get started now

You can get access to all the packages used in this book in < 1 minute, using this command in a terminal:

```
Listing 0.1 bash

docker run -it ghcr.io/js2264/biocbookdemo:devel R
```

Automated versioning of the online book

Regardless of whether the book package is submitted to Bioconductor, a Github Actions workflow publishes individual online books for each branch (named devel or RELEASE_X_Y) of a BiocBook-based Github repository.

For example, the online book version matching the devel version of the {BiocBook} package is available from:

http://js2264.github.io/BiocBookDemo/devel/

RStudio Server

An RStudio Server instance based on a specific Bioconductor <version> (devel or RELEASE_X_Y) can be initiated from the corresponding Docker image as follows:

Listing 0.2 bash

```
docker run \
    --volume <local_folder>:<destination_folder> \
    -e PASSWORD=OHCA \
    -p 8787:8787 \
    ghcr.io/<github_user>/<biocbook_repo>:<version>
```

The initiated RStudio Server instance will be available at https://localhost:8787.

Further instructions regarding Bioconductor-based Docker images are available here.

Acknowledgments

This works was inspired by and closely follows the strategy used in coordination by the Bioconductor core team and Aaron Lun to submit book-containing packages (from the OSCA series as well as SingleR and csaw books).

- @OSCA
- @SingleR
- @csaw

This package was also inspired by the *down package series, including:

- @knitr
- @bookdown
- @pkgdown

Session info

i Click to expand

References

Preamble

1 BiocBook versioning system

⚠ Important points

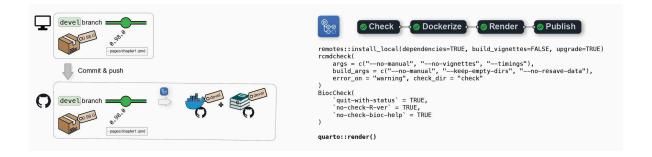
- Any package built using the {BiocBook} package is itself a {BiocBook}-based package;
- As such, it follows the release schedule from Bioconducor;
- The pages/ folder contains a number of pages which are rendered as a website using Quarto;
- The name of the branch (devel or RELEASE X Y) is crucial, as it is used to select a Bioconductor version to 1) build a **Docker image** for the {BiocBook}-based package and 2) render the BiocBook website.

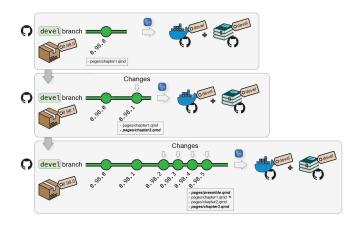
1.1 Continuous Integration and Continuous Delivery

1.1.1 From local to Github

1.

2.

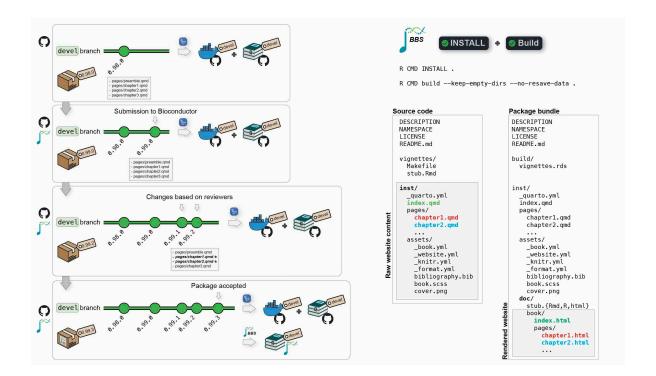




1.1.2 Package submission to Bioconductor

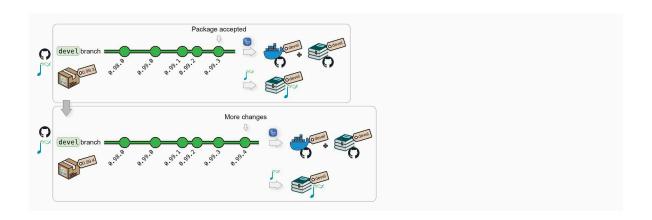
same reviewing process

1. 2.	Single Package Builder
3.	
4.	
5. 6. Build System	Bioconductor
7	



•

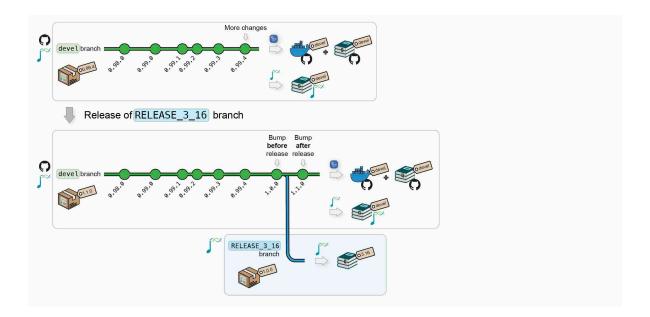
.



1.1.3 New Bioconductor releases

1.

2.



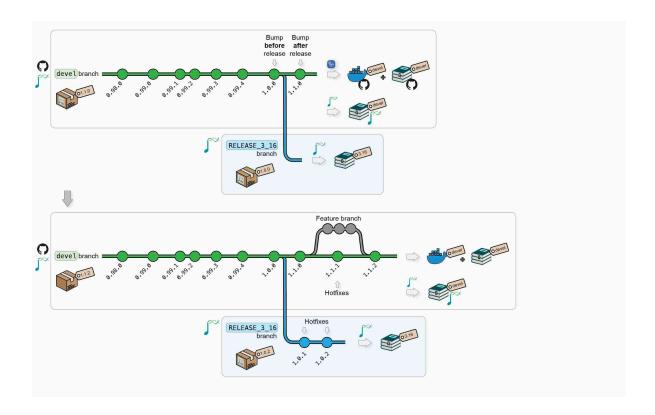
Note

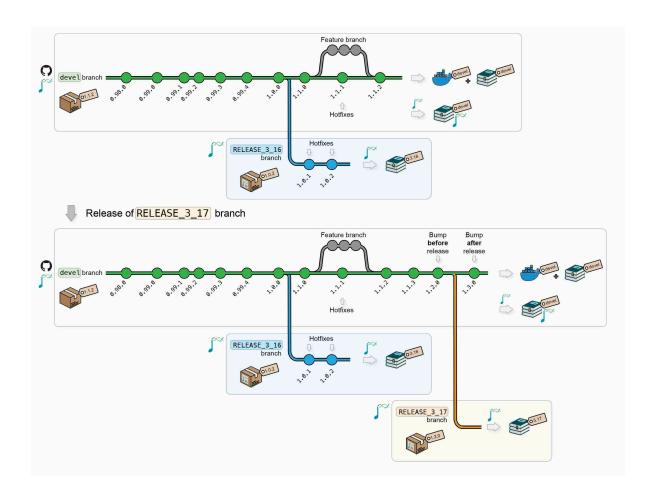
A {BiocBook}-based **package** can follow its own release life cycle if the autor does not intend to submit it to Bioconductor.

If the author of a {BiocBook}-based **package** intends to submit this package/book/website to Bioconductor, the Bioconductor submission and release life cycle:

- When developing a {BiocBook}-based **package** (at the submission and during review), the **package** version should be between 0.99.0 and 1.0.0.
- Once the submission is accepted and Bioconductor releases a new version X.Y, the {BiocBook}-based **package** version will automatically be bumped to 1.0.0 in Bioconductor release X.Y and to 1.1.0 in the continuing Bioconductor devel.

1.1.4 Updates





1.2 Access to versioned Docker and online book

1.2.1 Docker images versioning

- ghcr.io/js2264/biocbookdemo:devel
- ghcr.io/js2264/biocbookdemo:3.17
- \bullet ghcr.io/js2264/biocbookdemo:3.16
- \bullet ghcr.io/js2264/biocbookdemo:3.15

1.2.2 Website versioning

- https://js2264.github.io/BiocBookDemo/devel/
- https://js2264.github.io/BiocBookDemo/3.17/
- https://js2264.github.io/BiocBookDemo/3.16/
- https://js2264.github.io/BiocBookDemo/3.15/

1.3 Does this really work?

```
Sys.getenv("BIOCONDUCTOR_DOCKER_VERSION")
## [1] "3.20.16"
```



Note that this variable will always match the X.Y version returned by BiocVersion used to render the online book.

```
packageVersion("BiocVersion")
## [1] '3.20.0'
```

1.4 So what packages can I use?

```
packageVersion("BiocVersion")
## [1] '3.20.0'
BiocManager::available("BiocBaseUtils")
## [1] "BiocBaseUtils"
BiocManager::available("BiocHail")
## [1] "BiocHail"
BiocManager::available("CuratedAtlasQueryR")
## [1] "CuratedAtlasQueryR"
```

2 Writing a {BiocBook} package

2.1 Register a Github account in R

2.1.1 Creating a new Github token

```
usethis::create_github_token(
    description = "BiocBook",
    scopes = c("repo", "user:email", "workflow")
)
```

- •
- •
- •
- •

2.1.2 Register your new token in R

```
gitcreds::gitcreds_set()
```

Saving your Github token for later use

On Linux, gitcreds is generally not able to permanently store the provided Github token. For this reason, you may want to also add your Github token to ~/.Renviron to be able to reuse it. You can edit the ~/.Renviron by typing usethis::edit_r_environ(), and define the GITHUB_PAT environment variable:

```
Listing 2.1 .Renviron
```

```
GITHUB_PAT="<YOUR-TOKEN>"
```

2.1.3 Double check you are logged in

```
gh::gh_whoami()
```

2.2 BiocBook workflow

2.2.1 Initiate a {BiocBook} package

2.2.1.1 R

```
if (!require("BiocManager", quietly = TRUE)) install.packages("BiocManager")
if (!require("BiocBook", quietly = TRUE)) BiocManager::install("BiocBook")
library(BiocBook)
biocbook <- init("myBook")</pre>
```

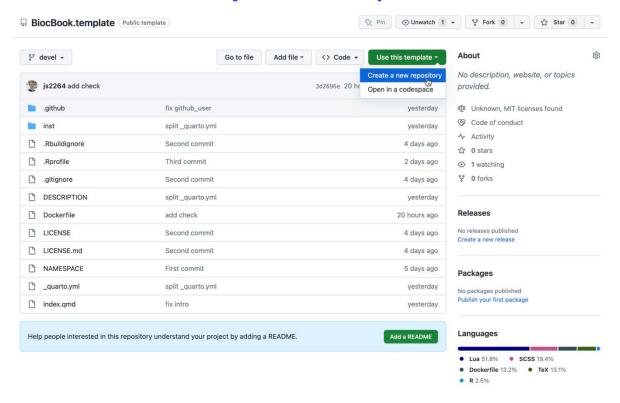
- 1.
- 2.
- 3.

2.2.1.2 VS Code

⚠ This approach is significantly more hazardous. It is highly recommended to stick to the init() helper function from the {BiocBook} package.

2.2.1.2.1 *

js2264/BiocBook.template

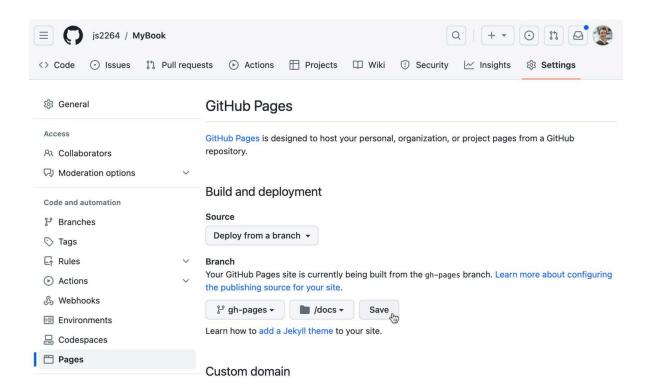


2.2.1.2.2 *

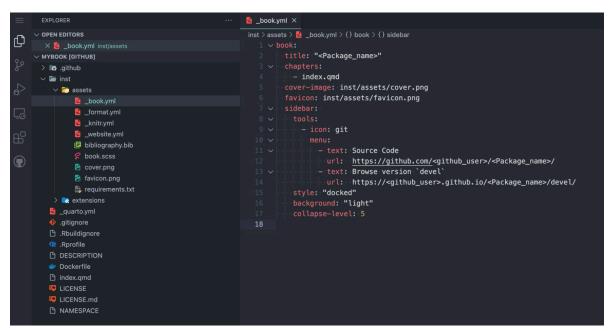
Create a new repository A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository. Required fields are marked with an asterisk (*). Repository template Start your repository with a template repository's contents. Include all branches Copy all branches from js2264/BiocBook.template and not just the default branch. Owner * Repository name * MyBook is available. Great repository names are short and memorable. Need inspiration? How about studious-guide? Description (optional) Public Anyone on the internet can see this repository. You choose who can commit. $\ensuremath{\ensuremath{\mathfrak{3}}}$ You are creating a public repository in your personal account. Create repository

2.2.1.2.3 Enable Github Pages to be deployed

- •
- •
- •
- •



2.2.1.2.4 *



2.2.1.2.5 *



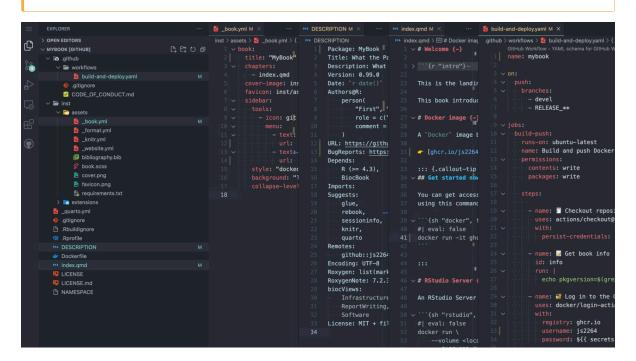
Warning

Three types of placeholders need to be replaced:

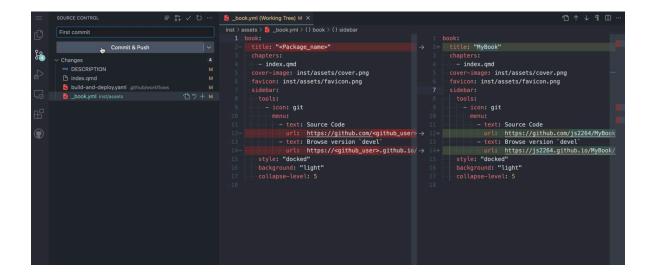
- 1. <Package_name>
- 2. <package_name>
- 3. <github_user>

Three different files contain these placeholders:

- 1. /inst/assets/_book.yml
- 2. /DESCRIPTION
- 3. /index.qmd



2.2.1.2.6 *



2.2.1.2.7 *

2.2.2 Edit new BiocBook chapters

•

•

⚠ Don't forget to add any package used in the book pages to Imports: or Suggests: fields in DESCRIPTION.

This ensures that these packages are installed in the Docker image prior to rendering.

2.2.3 Edit assets

- •
- •
- •

```
edit_yml(biocbook)
edit_bib(biocbook)
edit_css(biocbook)
```

2.2.4	Previewing	and	publishing	changes
-------	------------	-----	------------	---------

2.2.4.1 Previewing

preview(biocbook)

2.2.4.2 Publishing

publish(biocbook, message = " Publish")

2.2.4.3 Check your published book and Dockerfiles

status(biocbook)

2.3 Writing features

2.3.1 Executing code

2.3.1.1 R code

2.3.1.2 bash code

Listing $2.2\ R$

```
utils::packageVersion("BiocVersion")
## [1] '3.20.0'
```

Listing 2.3 bash

```
find ../ -name "*.qmd"
## ../index.qmd
## ../pages/Chapter-1.qmd
## ../pages/Chapter-2.qmd
## ../pages/preamble.qmd
## ../pages/Chapter-3.qmd
## ../pages/biocbook-vs-rebook.qmd
```

2.3.2 Creating data object

here

```
isthisworking <- "yes"
saveRDS(isthisworking, 'isthisworking.rds')</pre>
```

2.3.3 Adding references

3 Containerizing and Publishing against specific Bioconductor release versions

3.1 For a {BiocBook} package not currently in Bioconductor

3.2 For a {BiocBook} package accepted in Bioconductor > 6 months ago

here

gert::git_remote_add(name = 'upstream', url = 'git@git.bioconductor.org:packages/<YOUR-REPOS</pre>

```
gert::git_branch_create("RELEASE_3_15")

e
gert::git_pull("RELEASE_3_15", remote = "upstream")

•
```

gert::git_push("RELEASE_3_15", remote = "origin")

4 {BiocBook}-based books vs. {rebook}-based books

4.1 Differences with {rebook}-based books

Listing 4.1 R

```
work.dir <- rebook::bookCache('OSCA.intro')
handle <- rebook::preCompileBook('../inst/book', work.dir=work.dir, desc='../DESCRIPTION')
old.dir <- setwd(work.dir)
bookdown::render_book('index.Rmd')
setwd(old.dir)
rebook::postCompileBook(work.dir=work.dir, final.dir='../inst/doc/book', handle=handle)</pre>
```

Listing 4.2 shell

```
quarto render ../inst/
mv ../inst/docs ../inst/doc/book
```

 \triangle This implies that quarto (>= 1.3) has to be installed in the system building the package!

4.2 BiocBook features missing from rebook

- •
- •
- •
- •
- •

4.3 rebook features missing from BiocBook

•

? Tip

This can still be achieved within a book by saving a data object as an .rds file and loading it in a subsequent chapter.

```
isthisworking <- readRDS('isthisworking.rds')
isthisworking
## [1] "yes"</pre>
```

•