BIODIVERSITY BUILDING BLOCKS FOR POLICY

M8 Second quality assessment report of B3 software

23/02/2026

Author(s): Peter Desmet, Lissa Breugelmans, Shawn Dove, Katelyn Faulkner, Sanne Govaert, Quentin Groom, Sandra MacFadyen, Luisa Machado, Damiano Oldoni, Lien Reyserhove, Duccio Rocchini, Maxime Ryckewaert, Hanno Seebens, Maarten Trekels, Toon Van Daele, Mukhtar Yahaya

Funded by the European Union

This project receives funding from the European Union's Horizon Europe Research and Innovation Programme (ID No 101059592). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the EU nor the EC can be held responsible for them.

Table of contents

Summary

Software developed by B3 partners needs to be open, portable and of high quality. These intentions are outlined in the B-Cubed software development guide (Huybrechts et al. 2024), as 77 requirements that B3 software needs to meet. From December 3, 2025 to February 11, 2026, we reviewed twenty-six software tools developed by or related to B3 and assessed how well these are currently meeting the requirements.

Software tools meet the requirements overall (81%), especially regarding metadata and Python. None of the tools score below 57%. The ten software tools included in the first quality assessment (Desmet et al. 2024) improved their overall score from 76% to 82%. Most B3 tools are R packages (n = 12), which score well (83%). Almost all of these have tutorials that are included in the B-Cubed documentation website (https://docs.b-cubed.eu/) and form a “b3verse” of compatible packages (Langeraert et al. 2025).

While there remains room for improvement, the maintainers of B3 software clearly made sure their tools are open, portable and of high quality. This helps users and collaborators, and supports the technical sustainability goals (Depecker et al. 2025) of B3 outputs.

1. Introduction

Software is a crucial part of the B3 project. It is instrumental to many of the project deliverables and often a project deliverable itself, intended to be used by others, for different use cases and in different computing environments. That is why the software tools developed by B3 partners need to achieve a high level of quality, openness, and portability.

To aid software maintainers in accomplishing this, B3 published the **B-Cubed software development guide** in February 2024 (Huybrechts et al. 2024, initially created as deliverable D3.1). This document specifies high-level requirements for software, selected from numerous existing best practices and guidelines. It also provides hands-on instructions and examples. B3 partners and others have reported this guide to be useful in developing their software.

To assess how well B3 software currently meets the requirements laid out in the guide, we peer-reviewed the currently available tools. This quality assessment also serves as a follow-up of our first assessment in August 2024 (Desmet et al. 2024).

2. Methodology

We created an initial list of software to assess based on the first quality assessment (see review spreadsheet), the sustainability report (Depecker et al. 2025) and repositories that are part of the B3 organization on GitHub. Between October 31 and November 3, 2025, we contacted all maintainers (n = 21) of these software tools (n = 33) to indicate their software was considered for a high-level review. We asked whether they 1) wanted to exclude their software because it was out of scope (e.g. not related to B3 or its goals) or not suitable for review (e.g. development was abandoned or in very early stages), 2) were aware of other software tools we should consider, and 3) were willing to volunteer as a reviewer of software not their own. We also

informed them the review process would start in a couple of weeks, giving them an opportunity to make changes in alignment with the B-Cubed software development guide if desired.

Table 1: Software reviewed for this milestone. Software indicated with an asterisk was also included in the first quality assessment.

Name	Language	Type	Repository
b3alien	Python	package	https://github.com/mtrekels/b3alien
b3data-scripts	R	analysis	https://github.com/b-cubed-eu/b3data-scripts
b3doc	R	package	https://github.com/b-cubed-eu/b3doc
b3gbi*	R	package	https://github.com/b-cubed-eu/b3gbi
b3gbi-gui	R	shiny	https://github.com/b-cubed-eu/b3gbi-gui
binfrastructure*	Python	other	https://github.com/AgentschapPlantentuinMeise/binfrastructure
comp-unstructured-data	R	analysis	https://github.com/b-cubed-eu/comp-unstructured-data
dissmapr	R	package	https://github.com/b-cubed-eu/dissmapr
documentation	other	website	https://github.com/b-cubed-eu/documentation
dubicube	R	package	https://github.com/b-cubed-eu/dubicube
ebvcube*	R	package	https://github.com/EBVcube/ebvcube
flanders-use-case	R	analysis	https://github.com/b-cubed-eu/flanders-use-case
fowlplay*	Python	analysis	https://github.com/b-cubed-eu/fowlplay
gcube*	R	package	https://github.com/b-cubed-eu/gcube
impIndicator	R	package	https://github.com/b-cubed-eu/impIndicator
indicator-uncertainty	R	analysis	https://github.com/b-cubed-eu/indicator-uncertainty
indicators*	R	analysis	https://github.com/trias-project/indicators
integrate-indicator-software	R	analysis	https://github.com/b-cubed-eu/integrate-indicator-software
invasimapr	R	package	https://github.com/b-cubed-eu/invasimapr
occurrence-cube*	Java	other	https://github.com/gbif/occurrence-cube
pdindicatoR	R	package	https://github.com/b-cubed-eu/pdindicatoR
pygbif*	Python	package	https://github.com/gbif/pygbif
rgbif*	R	package	https://github.com/ropensci/rgbif
rsa-unstructured-data-comp	R	analysis	https://github.com/b-cubed-eu/rsa-unstructured-data-comp
trias*	R	package	https://github.com/trias-project/trias
vscube	R	package	https://github.com/b-cubed-eu/vscube

All maintainers agreed with this assessment, but six code repositories were considered out of scope (b-cubed-eu.r-universe.dev, b-cubed-workshops, biospace25-demo, euroraccoon, lps25-demo, malpolon, muddymetrics). No additional tools were suggested, resulting in 26 software tools and analyses in total (see Table 1). We assigned two reviewers to each new tool (n = 16) and one reviewer to each previously assessed tool (n = 10), resulting in 42 reviews overall.

Eleven of the 21 maintainers volunteered for review, joined by six other reviewers. During the review process, two reviewers withdrew, resulting in 17 reviewers overall (the authors of this report). They were assigned two to four tools to review, based on expertise and not being a contributor. Review instructions were sent on December 3 and review took place from **December 3, 2025 to February 11, 2026**. The results of the review were collected in a review spreadsheet.

The review process and scoring **was the same as in the first quality assessment**.

For each software, the reviewers assessed **77 requirements**, i.e. the requirements as defined in the B-Cubed software development guide. Each criterion is associated with a chapter in the guide, e.g. “An installable software tool MUST be maintained in its own repository.” is part of the “Code repositories” chapter. The reviewer had to assess each requirement and indicate a result (see Table 2), with an option to leave comments.

Table 2: Possible results for each requirement.

Name	Description
Complete	Software meets the requirement completely.
Not entirely	Software almost meets the requirement.
Lacking	Software does not meet the requirement.
Not applicable	The requirement does not apply for this software. For example “R code MUST be placed in the R/ directory of the repository.” does not apply for Python software.
Cannot assess	This requirement cannot be assessed for this software. This may indicate a poorly phrased requirement in the software development guide.
To assess	Requirement has not been assessed. A review is incomplete as long as there are requirements with this result.

Once reviews were completed, the first author of this report investigated the results for (obvious) discrepancies and wrong assessments, and notified reviewers if these were altered. Three scores were then calculated per software:

1. An **average count per result**, i.e. how often a certain result (e.g. “Complete”) was assigned to the software. This gives an indication how the reviewers assessed the software overall. For example, if reviewer 1 assessed the requirements as “Not entirely” three times and reviewer 2 two times, then the average count for “Not entirely” was 2.5.

2. An **average score per chapter in the guide**. This gives an indication how well the software scores for each chapter.

   First, a score was assigned per result per reviewer:
   - Completely: 100% (1.0)
   - Not entirely: 50% (0.5)
   - Lacking: 0% (0.0)
   - Not applicable: NA
   - Cannot assess: empty value

   Second, the scores were averaged for the two reviewers, where a numeric score by one reviewer trumps an NA or empty by the other reviewer.

   Finally, the scores were averaged by the number of applicable criteria per chapter. For example, reviewer 1 assigns NA, NA, 1.0 to the three criteria in the chapter “Tutorials”, while reviewer 2 assigns 1.0, 0.0, 1.0. NA values are ignored, so the average score is 1.0, 0.0, 1.0. Divided by the number of criteria for this chapter, the final score is 67% (2.0/3). When both reviewers assigned NA to all criteria in a chapter, the final score is NA.

3. A **total score**. This gives an indication how well the software scores overall. The same method is used as the average score per chapter in the guide, but across chapters (see Table 3).

3. Results

3.1 Review process

Reviewers reported “Cannot assess” very rarely (0.4%). This indicates that the requirements in the guide are well-defined and testable.

Reviewers also reported that the review process was informative for their own software development. Software review can thus act as a learning experience.

3.2 Overall results

Software received an **average total score of 81%**, with none below 57% and one (gcube) scoring 100% (see Table 3). This is an improvement over the first quality assessment, which had an average total score of 76%, with none below 50% and none reaching 100%. The ten tools that were reviewed as part of the first assessment increased their overall score from 76% to 82% and new tools started at a higher average baseline of 80%. This indicates that B3 software is **overall of high quality** and that this improves over time. The maintainers are clearly using the B-Cubed software development guide to follow best practices.

Table 3: Overall scores for the software reviewed for this milestone.

	Code repositories	The README file	Code collaboration	Versioning	R	R functions	R packages	R analysis code	Python	Tutorials	Total
b3alien	79%	100%	75%	100%	NA	NA	NA	NA	100%	67%	**88%**
b3data-scripts	100%	100%	100%	100%	63%	92%	NA	63%	NA	100%	**86%**
b3doc	100%	100%	81%	100%	95%	82%	80%	NA	NA	92%	**88%**
b3gbi	100%	83%	75%	100%	88%	72%	86%	NA	NA	100%	**84%**
b3gbi-gui	71%	100%	75%	50%	59%	94%	NA	92%	NA	25%	**74%**
binfrastructure	83%	83%	50%	0%	NA	NA	NA	NA	67%	0%	**57%**
comp-unstructured-data	96%	100%	94%	0%	50%	77%	NA	88%	NA	0%	**69%**
dissmapr	100%	92%	69%	100%	71%	70%	48%	NA	NA	67%	**70%**
documentation	100%	92%	56%	50%	NA	NA	NA	NA	NA	92%	**85%**
dubicube	100%	100%	100%	100%	95%	89%	100%	NA	NA	100%	**96%**
ebvcube	93%	100%	63%	75%	88%	90%	61%	NA	NA	100%	**82%**
flanders-use-case	100%	100%	63%	0%	54%	69%	NA	69%	NA	58%	**66%**
fowlplay	100%	100%	75%	100%	NA	NA	NA	NA	100%	67%	**93%**
gcube	100%	100%	100%	100%	100%	100%	100%	NA	NA	100%	**100%**
impIndicator	100%	100%	100%	100%	86%	79%	96%	NA	NA	92%	**91%**
indicator-uncertainty	96%	92%	81%	100%	68%	93%	NA	81%	NA	42%	**84%**
indicators	86%	100%	50%	67%	73%	68%	NA	63%	NA	67%	**71%**
integrate-indicator-software	100%	100%	75%	0%	67%	NA	NA	83%	NA	0%	**67%**
invasimapr	100%	100%	50%	100%	65%	82%	32%	NA	NA	0%	**64%**
occurrence-cube	92%	100%	25%	100%	NA	NA	NA	NA	NA	67%	**76%**
pdindicatoR	100%	100%	75%	75%	77%	98%	71%	NA	NA	75%	**84%**
pygbif	86%	100%	88%	100%	NA	NA	NA	NA	89%	33%	**84%**
rgbif	93%	100%	88%	100%	88%	94%	96%	NA	NA	83%	**93%**
rsa-unstructured-data-comp	100%	100%	100%	100%	100%	91%	NA	100%	NA	67%	**95%**
trias	93%	83%	75%	83%	92%	69%	79%	NA	NA	83%	**81%**
vscube	96%	100%	50%	0%	75%	77%	32%	NA	NA	100%	**66%**
total	**95%**	**97%**	**74%**	**73%**	**78%**	**83%**	**74%**	**80%**	**89%**	**64%**	**81%**

Chapters that score well were The README file (97%, n = 26), Code repositories (95%, n = 26), Python (89%, n = 4), R functions (83%, n = 19), R analysis code (80%, n = 8) and R (78%, n = 20). Improvements can be made in R packages (74%, n = 12), Code collaboration (74%, n = 26), and Versioning (73%, n = 26). As in the first quality assessment, the Tutorials requirement scored the lowest (64%, n = 26), which is expected because 1) tutorials are often only written when software is reaching a mature state and 2) the number of new software tools has increased. Software indicated with the repo status “Active” (as opposed to “Work in progress”) scores 71% for this chapter.

Most of B3 tools are R packages (see Table 1, n = 12), which score well (83%). Almost all of these have tutorials included in the B-Cubed documentation website (https://docs.b-cubed.eu/) and form a “b3verse” of compatible packages (Langeraert et al. 2025).

3.3 Results per software

See Table 3 for the scores per software. Software maintainers and contributors are welcome to consult the review spreadsheet, read the B-Cubed software development guide and/or contact the authors to see how their software can be improved.

3.4 Next steps

This is the last project-wide quality assessment of B3 software. We will now follow the sustainability report (Depecker et al. 2025) for long-term maintenance of B3 software. Maintainers are welcome to self-assess their software (or ask a colleague) following the methods described in this report.

4. Acknowledgements

We thank the software maintainers (and contributors) for developing and maintaining tools that are instrumental to the B3 project. It is not always easy to have your software scrutinized by others, so we thank you to be open for review: Matt Blissett (occurrence-cube), Lissa Breugelmans (pdindicatoR), Rocìo Beatriz Cortès (vscube), Peter Desmet (documentation), Shawn Dove (b3gbi, b3gbi-gui), Sanne Govaert (b3doc), Quentin Groom (fowlplay), Katelyn Faulkner (rsa-unstructured-data-comp), Jasmijn Hillaert (flanders-use-case), Ward Langeraert (b3data-scripts, comp-unstructured-data, dubicube, gcube, indicator-uncertainty, integrate-indicator-software), Sandra MacFadyen (dissmapr, invasimapr), Emmanuel Oceguera (ebvcube), Damiano Oldoni (indicators, trias), Maarten Trekels (b3alien), Christophe Van Neste (binfrastructure), John Waller (pygbif, rgbif) and Mukhtar Yahaya (impIndicator).

5. References

1. Depecker J, Abraham L, Pijls S, Trekels M, Yovcheva N, Groom Q (2025). Sustainability report. B3 project deliverable D1.8. https://b-cubed.eu/storage/app/uploads/public/68d/cbc/6b6/68dcbc6b6bb1d737971254.pdf
2. Desmet P, Breugelmans L, Dove S, Huybrechts P, Langeraert W, MacFadyen S, Oldoni D, Trekels M, Van Daele T, Van Neste C (2024). First quality assessment report of B3 software. B3 project milestone M7. https://b-cubed.eu/storage/app/uploads/public/677/e8b/a71/677e8ba71758d167483957.pdf
3. Huybrechts P, Trekels M, Abraham L, Desmet P (2024). B-Cubed software development guide. https://docs.b-cubed.eu/guides/software-development/
4. Langeraert W, Breugelmans L, Desmet P, Shawn D, Kumschick S, MacFadyen S, Seebens H, Trekels M, Yahaya M, Van Daele T (2025). Version 0.1 of the b3verse R package suite publicly available. B3 project milestone M29. https://b-cubed.eu/storage/app/uploads/public/68b/558/838/68b5588386b89344344899.pdf