



OPINION ARTICLE

Insights from the SwissRN Computational Reproducibility Hackathon 2025

[version 1; peer review: awaiting peer review]

Tom Willems ^{1,2}, Tobias Kühlwein ³, Matthias Voigt³, Michael Kurschilgen ³,
Daniel J Stekhoven ^{4,5}

¹Department of Psychology, University of Zurich, Zurich, 8050, Switzerland

²Institute of Psychology, University of Bern, Bern, 3012, Switzerland

³UniDistance Suisse, Brig, 3900, Switzerland

⁴NEXUS Personalized Health, ETH Zurich, Zurich, 8092, Switzerland

⁵Swiss Institute of Bioinformatics, Lausanne, 1015, Switzerland

V1 First published: 09 May 2026, 15:692
<https://doi.org/10.12688/f1000research.180013.1>
Latest published: 09 May 2026, 15:692
<https://doi.org/10.12688/f1000research.180013.1>

Open Peer Review

Approval Status AWAITING PEER REVIEW

Any reports and responses or comments on the article can be found at the end of the article.

Abstract

Computational reproducibility hackathons provide a hands-on opportunity for identifying barriers to reproducible research. The Swiss Reproducibility Network (SwissRN) hosted a Computational Reproducibility Hackathon at UniDistance Suisse in Brig, Switzerland, with participants from diverse scientific disciplines. 19 participants attempted to reproduce five published computational analyses, achieving at least partial success in four cases. Documentation quality, software environments, and data accessibility emerged as the most critical factors for successful reproduction. These findings inform ongoing efforts to develop practical best-practice guidance and training initiatives to improve computational reproducibility across disciplines.

Keywords

Reproducibility, Hackathon, Open Science, Metascience, Computational Research, Software, Community Engagement



This article is included in the **Research on Research, Policy & Culture** gateway.



This article is included in the **Hackathons** collection.

Corresponding author: Tom Willems (tomicwillems@pm.me)

Author roles: **Willems T:** Formal Analysis, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Kühlwein T:** Writing – Review & Editing; **Voigt M:** Conceptualization, Writing – Review & Editing; **Kurschilgen M:** Conceptualization, Writing – Review & Editing; **J Stekhoven D:** Conceptualization, Formal Analysis, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2026 Willems T *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Willems T, Kühlwein T, Voigt M *et al.* **Insights from the SwissRN Computational Reproducibility Hackathon 2025 [version 1; peer review: awaiting peer review]** F1000Research 2026, 15:692 <https://doi.org/10.12688/f1000research.180013.1>

First published: 09 May 2026, 15:692 <https://doi.org/10.12688/f1000research.180013.1>

Introduction

Reproducibility is fundamental to science because it ensures that research results are transparent and verifiable. Computational reproducibility focuses on the ability to obtain consistent results using the original data and computational steps, essentially testing the robustness of the code and analysis. Within the framework of open science, computational reproducibility is tightly connected to, yet distinct from, the broader concept of replicability, which addresses whether a similar result can be achieved using the same methods with new data.

At a reproducibility hackathon, the goal is to use the provided data and code, usually from a published article, to determine whether the analysis described by the authors can be repeated with consistent results. Besides confirming that the computations run as intended, this often involves checking compatibility across different operating systems and software versions. The process relies heavily on full access to both data and code, making open sharing essential. For software in particular, clear documentation and adherence to established standards and ontologies are equally important pillars of reproducible work.

Such hackathons offer excellent hands-on experience, especially for early-career researchers, who can develop good research habits before rigid routines form. Building on the success of the reproducibility hackathon satellite event at the Swiss Reproducibility Conference 2024, an independent hackathon was held at UniDistance Suisse in Brig, Switzerland on June 6, 2025. The event brought together 19 participants representing a vast range of scientific disciplines and eager to strengthen reproducible research practices. In this report, we present key insights that emerged during the event.

Results

Details of the reproduction efforts are available on ReproHack Hub (<https://www.reprohack.org/event/34/>), including the individual reproduction reports. Five original research papers were submitted for the hackathon, with some authors attending and engaging in reproducing work other than their own. The event emphasized a constructive and collaborative atmosphere, focusing on learning opportunities rather than criticism of the paper content. Authors who were present answered questions and provided support during the reproduction attempts.

Participants came from diverse scientific backgrounds, many were unfamiliar with the specific topics or computational methods involved. Working in groups and having direct access to the authors created a productive and supportive environment that enabled mutual learning for both participants and authors.

Four of the five submissions were at least partially reproduced (Table 1). Participants encountered several common hurdles, including unclear or missing documentation, unspecified software dependencies, and difficulties accessing code or datasets. Differences in hardware setups also contributed to variations in results, complicating replication efforts. One difficulty was the lack of version documentation for packages within the used programs. This led to the inability to reproduce certain aspects as the functionality of packages changed over time. Further, certain individualized variables were only partially explained which made it difficult to understand their exact usage and how to manipulate them. Throughout the event, participants tackled these issues and provided valuable feedback to the original authors. Many challenges appeared solvable by improving or adding metadata and clarifying documentation, while others, such as configuring containerized environments, required more technical expertise. At the end of the day, teams shared their findings and discussed obstacles and potential solutions, fostering a deeper understanding of how reproducibility can be improved in practice.

The mean reproducibility score (MRS) shown in Table 1 represents the average of all submitted reviews to the question: “How much of the paper did you manage to reproduce?”, rated on a scale of 0 to 10.

Participants shared their experiences as shown in Figure 1, expressing a mix of frustration and engagement throughout the reproduction efforts. Despite the difficulties, there was a clear interest and satisfaction in contributing to the important topic of reproducibility. Many appreciated the group setting and the exchange of perspectives across different levels of expertise.

Large language models (LLMs), such as ChatGPT, played a helpful role in creating computational environments even for those with limited technical knowledge. Outdated software and package versions proved to be significant obstacles, alongside insufficient documentation. While researchers have limited control over software stability, the responsibility for clear documentation lies with the authors. However, the lack of strong incentives often discourages investment in detailed documentation. LLMs may help ease this burden by improving clarity and readability, although their use should remain cautious and supervised. Participants later emphasized, as reflected in Figure 2, that effective documentation is one of the most critical factors for successful reproducibility.

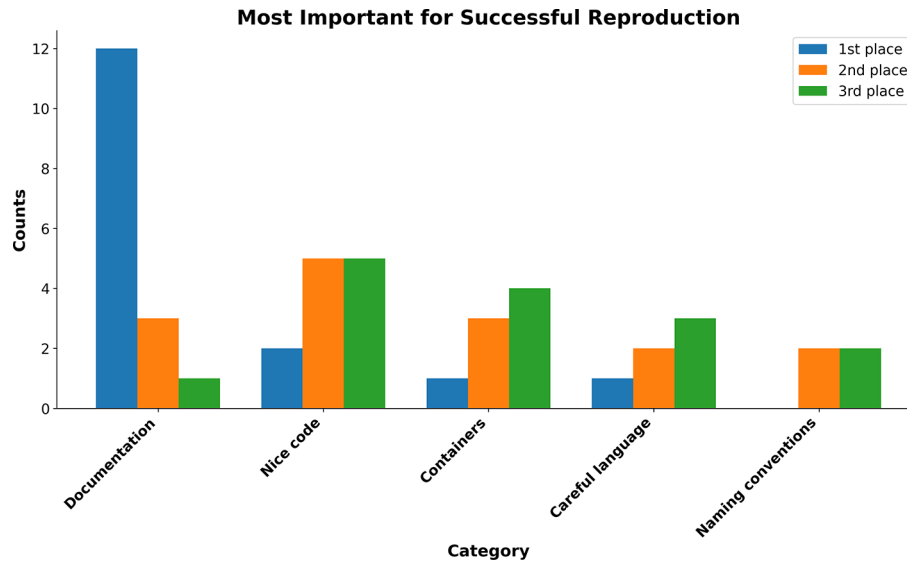


Figure 2. Summary of participant responses on factors enabling reproducibility.

Discussion

The discussions at the end of the hackathon converged on a shared understanding that computational reproducibility is less a single technical hurdle and more an ecosystem problem, shaped by documentation quality, software evolution, and community norms. While participants encountered a wide range of domain-specific challenges, the obstacles they identified were remarkably consistent across projects and disciplines.

A central theme was the importance of clear, structured documentation, which participants ranked as the single most important factor when attempting to reproduce computational work. Documentation was repeatedly described as the primary entry point into a project, enabling others to understand assumptions, variable definitions, data preprocessing steps, and expected outputs. Even well-written and logically structured code proved difficult to reuse in the absence of sufficient explanatory context. This aligns with the observation that documentation often determines whether reproduction efforts can begin at all, rather than how efficiently they proceed.

Closely related to documentation, environment capture and documentation emerged as a top priority in the participant-derived checklist, receiving the highest importance rating (mean 4.4 on a 1–5 scale, tied with data accessibility and versioning). Participants emphasized that reproducibility is strongly undermined when software dependencies, package versions, or operating system assumptions are not explicitly recorded. The rapid evolution of scientific software ecosystems means that even relatively recent code may fail to execute without careful environment specification. While researchers cannot fully control external software changes, they can mitigate their impact through explicit version pinning, environment files, and containerized setups.

Data accessibility and versioning were rated equally important (mean 4.4), reflecting the practical reality that reproduction efforts cannot proceed if datasets are unavailable, poorly documented, or ambiguously versioned. Participants noted that even when data were technically accessible, missing metadata, unclear preprocessing steps, or undocumented transformations created substantial barriers. Persistent identifiers for datasets and clear links between data versions and specific analyses were repeatedly highlighted as good practice.

Other checklist components (code structure, sharing, and execution (mean 3.9), result provenance and reporting (mean 3.8), and randomness control and determinism (mean 3.3)) were seen as important but secondary. This ordering suggests a pragmatic perspective: reproducibility first depends on being able to run an analysis at all, before finer-grained concerns such as numerical determinism or formal provenance tracking become relevant. Participants also ranked “nice code” and “containers” highly when asked about the most important aspects of reproduction, reinforcing the idea that readability, modularity, and standardized execution environments meaningfully lower the barrier for third-party reuse.

Importantly, these discussions underscored that reproducibility should be approached as a process designed from the outset, rather than a retrospective fix. Participants agreed that early planning – such as deciding where code and data will

be hosted, how versions will be tracked, and how results will be reported – has a disproportionately large impact on downstream reproducibility. Rather than debating optimal tools, participants advocated for clear, well-documented choices that can be understood and reused by others. Support structures, including research software engineers and data stewards, were identified as key enablers, particularly for researchers without formal training in software development.

Finally, the role of institutional and journal-level policies was discussed. Mandatory data and code sharing requirements were widely acknowledged as having already improved reproducibility standards, although participants noted that compliance alone does not guarantee usability. In this context, lightweight tools, such as checklists, example repositories, and shared templates, were seen as promising mechanisms to translate policy requirements into meaningful practice.

Outlook

The SwissRN Computational Reproducibility Hackathon demonstrated that hands-on reproduction efforts can generate not only valuable feedback for individual projects, but also transferable insights into the structural conditions that enable or hinder computational reproducibility. Building on these experiences, a central ambition emerging from this initiative is to move beyond isolated events toward the development of widely accessible, community-driven best practices for computational reproducibility, and to embed these practices sustainably into research training.

Repeated hackathons conducted across different regions, institutions, and scientific domains, without being fixed to a specific topic, provide a particularly powerful mechanism for identifying generalizable reproducibility principles. By deliberately exposing a wide variety of computational workflows to reproduction attempts, such events allow recurring challenges and effective solutions to surface independently of discipline-specific conventions. At the same time, they naturally reveal topical or methodological special cases, which can be documented as extensions rather than exceptions to a shared core of best practices.

A key next step is therefore to distill the insights gained from multiple hackathons into a concise, practical best-practice guide for computational reproducibility, grounded in empirical experience rather than abstract recommendations. Participant feedback from this and future events can directly inform the structure of such guidance, prioritizing aspects that have the greatest downstream impact, such as documentation quality, environment capture, and data accessibility, while offering pragmatic advice on code organization, provenance tracking, and randomness control. Making this guidance openly available in modular, reusable formats (e.g., checklists, templates, example repositories) will be essential to ensure broad adoption.

Beyond documentation, there is a clear opportunity and responsibility to translate these best practices into effective educational offerings, particularly at the undergraduate level and across disciplines. Teaching computational reproducibility early, before ad-hoc habits become entrenched, has the potential to raise baseline standards across entire research communities. Rather than treating reproducibility as an advanced or optional topic, it should be integrated into core curricula wherever computational methods are used.

To scale such efforts sustainably, we see strong alignment with established training frameworks such as The Carpentries,^{6,7} emphasizing reusable lesson materials, instructor training, and a “teach the teachers” model. Casting reproducibility content into this framework would allow best practices to be disseminated efficiently, adapted locally, and maintained collaboratively. In this context, close collaboration with the SwissRN Working Group for Training represents a natural and important next step, ensuring that methodological insights from hackathons are translated into pedagogically sound, high-quality training resources.

Ultimately, reproducibility hackathons should be understood as catalysts within a larger ecosystem; linking empirical assessment, community knowledge generation, and education. By systematically connecting hackathon-derived insights with open best-practice guidance and scalable training initiatives, SwissRN can contribute to a durable improvement in computational reproducibility across disciplines and career stages.

Data availability

Underlying data

Repository name: Insights from the SwissRN Computational Reproducibility Hackathon 2025. <https://doi.org/10.17605/OSF.IO/3WFSQ>.⁸

The project contains the following underlying data:

- `mentimeter_results.xlsx` (raw, unaveraged behavioral data)
- `create_figs.py` (script that uses behavioral data to create figures 1 and 2 for the manuscript)
- `figure1.tiff` (figure 1 as it is displayed in this manuscript)
- `figure2.tiff` (figure 2 as it is displayed in this manuscript)

Extended data

We do not report any extended data.

Data are available under the terms of the [Creative Commons Attribution 4.0 International License \(CC BY 4.0, https://creativecommons.org/licenses/by/4.0/\)](https://creativecommons.org/licenses/by/4.0/).

Acknowledgements

The authors would like to thank all participants of the hackathon: Meret Hildebrandt, Simon Ruch, Chhavi Sachdeva, Lucia-Manuela Cantonas, Emilie Morgan de Paula, Flora Logoz, Niels Kempkens, Alexandra Lapteva, Nadia Maggetti, Antoine Buetti-Dinh, Ivan Topolsky, Gabe Winter, Viktoriia Apalkova, Rimaite Auguste, Yulia Kulagina, Jelena Čuklina, Benjamin Dominitz, Tuba Kadriye. The authors would like to thank UniDistance Suisse for hosting the event.

References

1. Benner P, Himpe C, Mitchell T: **On reduced input-output dynamic mode decomposition**. *Adv Comput Math*. 2018 Dec; **44**(6): 1751–1768.
[Publisher Full Text](#)
2. Lapteva A, Schnyder S, Wolff W, et al.: **Bore me (not): boredom impairs recognition memory but not the pupil old/new effect**. *Q J Exp Psychol (Hove)*. 2025 Mar 13; **17470218251329255**: 17470218251329255.
[Publisher Full Text](#)
3. Pawel S, Held L: **Closed-form power and sample size calculations for Bayes factors**. *Am Stat*. 2025 Jul 3; **79**(3): 330–344.
[Publisher Full Text](#)
4. *gender-equity-navigator*. Github; [cited 2025 Nov 26].
[Reference Source](#)
5. Bagutti C, Alt Hug M, Heim P, et al.: **Wastewater monitoring of SARS-CoV-2 shows high correlation with COVID-19 case numbers and allowed early detection of the first confirmed B.1.1.529 infection in Switzerland: results of an observational surveillance study**. *Swiss Med Wkly*. 2022 Jun 20; **152**(2526): w30202.
[Publisher Full Text](#)
6. Wilson G: **Software Carpentry: lessons learned**. *F1000Res*. 2014 Feb 19; **3**(62): 62.
[Publisher Full Text](#)
7. *The Carpentries*. The Carpentries; 2023 [cited 2026 Apr 10].
[Reference Source](#)
8. Willems TE, Stekhoven D, Kurschilgen M, et al.: **Insights from the SwissRN Computational Reproducibility Hackathon 2025**. *OSF*. 2026 [cited 2026 Apr 21].
[Publisher Full Text](#)

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research