

January 30, 2024

Office of Data Science Strategy
National Institutes of Health
9000 Rockville Pike,
Bethesda, MD
20892.

RE: Request for Information (RFI): Best Practices for Sharing NIH Supported Research Software

Dear Office of Data Science Strategy,

Thank you for the opportunity to participate in this Request for Information (RFI) on Best Practices for Sharing NIH Supported Research Software. Given the National Institutes of Health's role as a major funder of scientific research in the U.S., we believe that guidance adopted by the NIH has the potential to guide best practices in scientific research on the sharing of research software. AGU's response to the RFI is based on our experience working with Earth, space, and environmental science researchers as well as computer and information scientists and software experts in the work that we are doing promoting FAIR and open software sharing in AGU journals and in the wider community. These efforts include providing guidance on best practices for software sharing upon publication, guidance for our researchers on enhancing the reusability of their research software, and working with AGU authors and editors to find appropriate software repositories. Our response to this RFI incorporates this experience with researchers.

Sincerely,



Kristina Vrouwenvelder
Program Manager,
AGU Open Science Leadership
ORCID: 0000-0002-5862-2502



Shelley Stall
Vice President,
AGU Open Science Leadership
ORCID: 0000-0003-2926-8353

1. Comment on the current NIH [Best Practices for Sharing Research Software](#).

The current NIH Best Practices for Sharing Research Software are set up as a FAQs document for researchers. We would recommend also sharing best practices in a format that indicates at which part of the research workflow each issue arises. For example, some elements of best practice of sharing research software will arise when the software is being developed and or the research is conducted, before any research outputs are shared. Specific comments associated with the FAQs are detailed below.

1. Why should I share software and code as “open source” software?

This FAQ details the benefit of sharing software, in particular the transparency that working openly enables. We recommend adding an emphasis on other benefits that sharing software provide to the researcher. For instance, sharing software makes research more reusable, reducing duplication of effort in research, increasing research efficiency, and allowing others to build off a research project. This reuse in turn can enhance a researcher’s impact and profile in the field, when paired with proper credit and attribution for software reuse. Research teams concerned about sharing software at early stages of research can restrict software sharing to the immediate research team and share software more broadly at a later time. Overall, sharing software helps boost the replicability of research; research software should be considered a vital part of the methods underlying a research paper as well as a valuable research output in its own right.

2. How do I make software source code ‘open’?

This FAQ describes the sharing of software through development platforms enabling version control and through preservation repositories. We recommend defining when software should be shared in the different formats (*‘as source code or the executable version, as code libraries published to general or specific package distribution channels, as workflows or containers, as services and APIs’*) described in the FAQ response. For instance, sharing software workflows is useful when research processes can’t be fully coded or automated, or for research employing proprietary software. Moreover, guidance should also be provided on when to share software outputs in specific cases, for instance model runs or simulation data. The EarthCube RCN, “What About Model Data?”, provides a useful rubric and other guidance on this: <https://modeldata.rcn.github.io/>. New formats for disseminating the results of scientific research, such as peer-reviewed computational notebooks, will serve as another useful format for sharing code and software that underlies research, and will greatly enhance the reproducibility and transparency of analysis (see AGU’s Notebooks Now project for our plans to pilot peer-reviewed computational notebook publication in our journals: <https://data.agu.org/notebooks-now/>). Finally, while this section advises that researchers choose a funder specified or preferred repository, in practice few funders provide these recommendations (and an NIH recommendation is not given here. If NIH as a funder has a preferred repository, best to specify here. More broadly, we recommend that funders work together with other

stakeholders (researchers, repositories) to identify community-accepted, trusted repositories for software sharing. GitHub and other collaborative development platforms are not repositories and do not offer persistent identifiers such as DOIs. This distinction may not be clear to researchers reading the current version of the FAQ. We recommend specifying this to researchers and recommending a combination of a collaborative development platform and repository providing long-term preservation and a persistent identifier (ex. GitHub + Zenodo).

3. *Why should I use a license when distributing code?*

We recommend specifying that researchers should use a license when distributing code to enable legal reuse of their code. Also, researchers should be guided to software-specific licenses, as employing licenses specific to text is a common pitfall we see with AGU researchers. For platforms where both code and text appear, licenses for both must be articulated.

4. *How do I choose a license under which to release software developed as part of an NIH award?*

No comment.

5. *How can I make my software citable?*

As stated above, we recommend (i) specifying to researchers the difference between code sharing on a collaborative development platform and code preservation in a trusted repository with a persistent identifier, and (ii) that funders work together with the community to identify and sustain trusted repositories. In AGU's 24 journals, we require authors to share information about the software that underlies their peer-reviewed research articles in the Availability Statement section of each research article, and if possible, that the software is shared in a community-accepted, trusted repository and given a DOI, so that a citation to the software can be included in the References section. Code should be available to reviewers upon first revision of the research article, enabling greater transparency of the work under review. We recommend that funders promote the practice of citing software in research publications to authors and work with other publishers to implement this practice in journals.

No comment on FAQs 6-9.

10. *What metadata should be considered when sharing research software?*

Metadata for software should adhere to community standards and should be machine readable. Funders should work with the community (including researchers and repositories) to develop and implement metadata standards, to enhance the findability, interoperability, and reusability of research software.

2. Describe how, when, and where you share your research software. What, if any, resources for best practices do you rely upon to make your shared software open and reusable?

As described above, AGU requires authors to share the software/code underlying their research article at the revisions stage of the paper and upon publication. Code must be provided to reviewers and software underlying the paper should be described in the Availability Statement section. We require authors to deposit software and code in a community-accepted, trusted repository – usually Zenodo through a GitHub link – when possible and require them to cite that software in the References section, if the software has a DOI. We recommend that funders work with researchers to encourage researchers to share their software with a DOI, and work with publishers to widely implement the sharing of research software alongside research articles and the citation of relevant software in research articles.

3. What existing standards or criteria do you use to evaluate the openness, FAIRness, quality, and/or security of the software you share or reuse?

No response.

4. Describe the collaborative settings in which you develop and share research software. Name communities or organizations, if any, you participate in that are actively promoting or developing software sharing best practices.

SciCodes, a consortium of over 35 scientific software registries and repositories, has released best practices for software sharing. (<https://scicodes.net/best-practices-for-software-registries-and-repositories/>) The Research Data Alliance Working Group FAIR Principles for Research Software (FAIR4RS) has released a recommendation for FAIR research software (<https://doi.org/10.15497/RDA00068>).

The FORCE11 Software Citation Implementation Working Group has developed:

Software citation principles:

Smith AM, Katz DS, Niemeyer KE, FORCE11 Software Citation Working Group. 2016. Software citation principles. *PeerJ Computer Science* 2:e86 <https://doi.org/10.7717/peerj-cs.86>

Software citation implementation challenges:

D. S. Katz, D. Bouquin, N. P. Chue Hong, J. Hausman, C. Jones, D. Chivvis, T. Clark, M. Crosas, S. Druskat, M. Fenner, T. Gillespie, A. Gonzalez-Beltran, M. Gruenpeter, T. Habermann, R. Haines, M. Harrison, E. Henneken, L. Hwang, M. B. Jones, A. A. Kelly, D. N. Kennedy, K. Leinweber, F. Rios, C. B. Robinson, I. Todorov, M. Wu, Q. Zhang, "Software Citation Implementation Challenges", arXiv 1905.08674 [cs.CY], 2019.

Checklists for (paper) authors and (software) developers:

N. P. Chue Hong, A. Allen, A. Gonzalez-Beltran, A. de Waard, A. M. Smith, C. Robinson, C. Jones, D. Bouquin, D. S. Katz, D. Kennedy, G. Ryder, J. Hausman, L. Hwang, M. B. Jones, M. Harrison, M. Crosas, M. Wu, P. Löwe, R. Haines, S. Edmunds, S. Stall, S. Swaminathan, S. Druskat, T. Crick, T. Morrell, T. Pollard, "Software Citation Checklist for Authors," Zenodo, 15-Oct-2019.

<https://doi.org/10.5281/zenodo.3479198>

N. P. Chue Hong, A. Allen, A. Gonzalez-Beltran, A. de Waard, A. M. Smith, C. Robinson, C. Jones, D. Bouquin, D. S. Katz, D. Kennedy, G. Ryder, J. Hausman, L. Hwang, M. B. Jones, M. Harrison, M. Crosas, M. Wu, P. Löwe, R. Haines, S. Edmunds, S. Stall, S. Swaminathan, S. Druskat, T. Crick, T. Morrell, T. Pollard, "Software Citation Checklist for Developers," Zenodo, 15-Oct-2019.

<https://doi.org/10.5281/zenodo.3482768>

Best practices for software repositories and registries:

Task Force on Best Practices for Software Registries: A. Monteil, A. Gonzalez-Beltran, A. Ioannidis, A. Allen, A. Lee, A. Bandrowski, B. E. Wilson, B. Mecum, C. Fan Du, C. Robinson, D. Garijo, D. S. Katz, D. Long, G. Milliken, H. Ménager, J. Hausman, J. H. Spaaks, K. Fenlon, K. Vanderbilt, L. Hwang, L. Davis, M. Fenner, M. R. Crusoe, M. Hucka, M. Wu, N. Chue Hong, P. Teuben, S. Stall, S. Druskat, T. Carnevale, T. Morrell, "Nine Best Practices for Research Software Registries and Repositories: A Concise Guide," arXiv 2012.13117 [cs.DL], 2020.

Guidance for journals:

D. S. Katz, N. P. Chue Hong, T. Clark, A. Muench, S. Stall, D. Bouquin, M. Cannon, S. Edmunds, T. Faez, P. Feeney, M. Fenner, M. Friedman, G. Grenier, M. Harrison, J. Heber, A. Leary, C. MacCallum, H. Murray, E. Pastrana, K. Perry, D. Schuster, M. Stockhause, J. Yeston, "Recognizing the value of software: a software citation guide [version 2; peer review: 2 approved]," F1000Research 9:1257, 2021. <https://doi.org/10.12688/f1000research.26932.2>

D. S. Katz, H. Murray, "Guest Post — Citing Software in Scholarly Publishing to Improve Reproducibility, Reuse, and Credit," Scholarly Kitchen, 21-Jan-2021.

S. Stall, G. Bilder, M. Cannon, N. Chue Hong, S. Edmunds, C. C. Erdmann, M. Evans, R. Farmer, P. Feeney, M. Friedman, M. Giampoala, R. B. Hanson, M. Harrison, D. Karaiskos, D. S. Katz, V. Letizia, V. Lizzi, C. MacCallum, A. Muench, K. Perry, H. Ratner, U. Schindler, B. Sedora, M. Stockhause, R. Townsend, J. Yeston, T. Clark, "Journal Production Guidance for Software and Data Citations," Scientific Data v.10, 656, 2023. <https://doi.org/10.1038/s41597-023-02491-7>

5. What factors influence your decision to share or reuse your research software (or not)? What technical, policy, financial, institutional, and/or social barriers to sharing or reuse of research software have you encountered?

We require AGU authors to share the research software and code that underlies their publications, preferably with a DOI and citation to that software as preserved in a repository, as previously described. However, we have seen various barriers to sharing software alongside research articles. These range from the social/cultural (concerns about scooping, inappropriate reuse, and even lack of polish on code and code documentation) to financial (lack of time and team expertise to properly document and share software for reusability). We believe that to enable a culture of software sharing, incentives are needed for researchers – which may range from credit for reuse of software enabled by software citation to open science awards like AGU's [Open Science Recognition Prize](#) or other incentives from funders. Education and guidance for researchers is also needed; this should be embedded into researcher training and the research workflow, including in the award process (analogously to Data Management Plans). Publishers and funders should also promulgate policies of software sharing that researchers must comply with.

6. Comment on your ability to reuse open-source research software developed by others. Describe factors used to determine whether to reuse existing research software or develop anew.

AGU researchers commonly cite issues related to the FAIR-ness of research software as barriers to reuse of software. These include the difficulties of finding open-source-research software for the question at hand; lack of proper documentation of existing research software, or other technical issues arising from sharing that software without computing environment information, etc. Many researchers feel that code underlying most research, excepting software packages quite deliberately developed by researcher/software developer teams for the sole purpose of reuse by the scientific community, is usually not shared to a FAIR standard enabling reuse without extensive assistance, to the extent of authorship on the new project, of the original creator.

7. How can NIH support research software communities of practice to better aid development of best practices for sharing and reuse of high-quality research software?

- Funders should provide sustainable funding for open-source scientific software development and for the embedding of research software engineers and other software experts on research teams
- Support for education and training of researchers on best practices for software development and sharing is needed

- Funders should support software sharing alongside the publication of primary research articles, and citation in those articles of the software underlying the research
 - To ease the adoption of FAIR practices on researchers and to enhance standardization and interoperability, support for FAIR automation and tools is needed
- 8. Comment on any other topic which may be relevant for NIH to consider in enhancing the sharing of research software.**

No response