Collaborating on Software Archiving for Institutions

Talya Cooper
New York University

Vicky Rampin
New York University

Software and code belong to our scholarly record. Scholars across disciplines use and create software; consequently, accessing software is key to reproducing research results. As open research practices become the norm in academia, due in part to the widespread acknowledgement of the "reproducibility crisis" and broad adaptation of the FAIR Principles, more researchers have begun sharing articles and data in open repositories designed for long-term preservation. However, for scholarly code, sharing often begins and ends on Git-hosted platforms (GHPs) such as GitHub, GitLab, or Bitbucket, where the code is typically updated (sometimes by multiple authors) and augmented with contextual information (such as discussion threads, wikis, web manuals, forums, project boards, etc.) that provides valuable information about the code's development and function.

The dearth of preservation solutions for code and its documentation–especially considering the wide range of specialist, disciplinary, and university-based repositories designed for long-term preservation of data– proves problematic when trying to reproduce scholars' work. Someone attempting to reproduce research needs the data; the code or software used to produce results; documentation about how to replicate computational workflows (including instructions about how to run the software, and the software dependencies it requires); and the ability to recreate the computing environment in which that software originally ran. While data may be available in the long-term via a repository, and accurate metadata (alongside a range of containerization and emulation tools) can enable recreation of a computing environment, code is typically tied to Web hosting platforms. But these platforms make no guarantees about their ongoing availability or the long-term preservation of the code they host. In the recent past, GHPs have been discontinued and taken offline (such as Google Code, which shut down in 2016); changed their URL structure, leading to link rot and potentially to data loss (Escamilla, 2022, March 30); and caused international researchers to lose access to their own projects due to sanctions (Friedman, 2021). In order to maintain computational projects' reproducibility, scholars must be able to preserve their code in stable repositories, outside the confines of these platforms.

Miliken's (2021) work identified four potential routes available for scholars who hope to archive their code: self-archiving in a repository like Zenodo, relying on programmatic capture by an indexing or crawling effort (such as the Software Heritage Foundation), using web archiving tools such as Webrecorder, and leveraging tools created for long-term preservation of legacy software. Currently, no code-specific repositories archive the important web-based context alongside the code. Moreover, Steeves, Miliken, and Nguyen (2021) found that only 47.5% of scholars who write code chose to self archive. We perceived a need for both advocacy and technical work around preservation best practices for research software.

Talya Cooper, tc3602@nyu.edu, 70 Washington Sq. South, #517, New York, NY 10012
Vicky Rampin, vs77@nyu.edu, 70 Washington Sq. South, #517, New York, NY 10012

Our current project, *Collaborating on Software Archiving for Institutions* (CoSAI), a multi-institutional collaboration funded by the Alfred P. Sloan Foundation, addresses this gap as we work to preserve web-based scholarship, scholarly code in particular. CoSAI's goal is to create reproducible, machine-repeatable, and human-understandable workflows for computational scholarship, while lowering the barriers to entry for software archiving. Through our project, we hope to promote education, outreach, and community building around this issue, first within our community of data curators and librarians, and ultimately to scholars, as well.

The project has three main areas of focus: 1) technical development of an open-source toolkit for selecting, describing, and accessioning open scholarship on the Web (with a focus on research software), 2) community building, education, and outreach around sustainability and reusability for research software, and 3) sharing workflows (based on the toolkit) for archiving open scholarship with its important scholarly ephemera (e.g., for software: issue threads, wikis, manuals).

Our idea is to link and extend data curators' work of ensuring that data can be understood and opened in the long-term with software preservation efforts.The tool we are developing combines the programmatic web crawler Memento Tracer with OCCAM, a powerful federated software preservation system. Using techniques from both the web archiving and software preservation communities, we capture software within its context from web-based platforms. Then, we package the software with the contextual material (stored as a WARC file) and descriptive and technical metadata. This package can subsequently be deposited into a preservation repository, where it can be linked to any related materials (e.g., related data or a scholarly article describing the project), and ultimately be runnable in a replay system, such as Emulation as a Service Infrastructure.

Eight months into the project, we have reached several key milestones. At NYU, we have created a software preservation scoping policy which outlines the types of research software that fall in scope for our work and the minimal necessary components with which they must be archived. We have also developed a metadata schema that uses the CodeMeta standard, with a crosswalk to the DataCite standard used by our institutional repository. We designed several workshops for the University community about designing research software for reproducibility and preservability, which we will pilot in Summer 2022. Additionally, technical development of the toolkit and workflows has begun, in collaboration with partners at Los Alamos National Laboratory, the developers of Memento Tracer, and University of Pittsburgh, where OCCAM development takes place. Finally, a team at Old Dominion University is conducting research to understand the scope of research software in existing scholarship, providing pathways for identifying research software at scale. Among their other findings, they have undertaken an analysis that shows that 1 in 5 articles published on arXiv in 2021 included a link to a GitHub repository (Escamilla, 2022, May 24)–an observation that points to software's ubiquity in contemporary scholarship, and emphasizes the need for active, in-depth research software preservation.

References

Escamilla, E. (2022, March 30). *GitHub Is Not An Archive - GitHub Pages*. Old Dominion Web Science and Digital Libraries Research Group. https://ws-dl.blogspot.com/2022/03/2022-03-30-github-is-not-archive-github.html

Escamilla, E. (2022, May 24). "*Archiving source code in scholarly content: One in five articles references GitHub*." [Lightning talk]. IIPC/WAC 2022, Virtual. https://docs.google.com/presentation/d/15yrKbnjalbbtt-FNtU2lp7IEfNuvEtXAP8PALUueUlM/present

Friedman, N. (2021, January 5). *Advancing developer freedom: GitHub is fully available in Iran*. GitHub Blog. https://github.blog/2021-01-05-advancing-developer-freedom-github-is-fully-available-in-iran/

Miliken, G. (2021). "IASGE Environmental Scan." [Grant report]. https://osf.io/ku24q.

Steeves, V., Miliken, G., & Nguyen, S. (2021). "A Behavioral Approach to Understanding the Git Experience." *HICSS* 54. https://doi.org/10.24251/HICSS.2021.872.