Software Curation in Research Libraries: Practice and Promise

Alexandra Chassanoff[1], John Borghi[2], Yasmin AlNoamany[3], Katherine Thornton[4]

1. Massachusetts Institute of Technology
2. California Digital Library
3. University of California, Berkeley
4. Yale University

Corresponding author: amchassanoff@gmail.com

Acknowledgements: From 2016 to 2018, the authors were Postdoctoral Fellows in Software Curation at four host institutions. This research was undertaken during that time period and funded in part by Council on Library and Information Resources (CLIR) and the Sloan Foundation.

This preprint is currently under review at *The Journal of Librarianship and Scholarly Communication.*

INTRODUCTION. Research software plays an increasingly vital role in the scholarly record. Academic research libraries are in the early stages of exploring strategies for curating and preserving research software, aiming to provide long-term access and use. **DESCRIPTION OF PROGRAM.** In 2016, the Council on Library and Information Resources (CLIR) began offering postdoctoral fellowships in software curation. Four institutions hosted the initial cohort of software curation fellows. This article describes the work activities and research program of the cohort, highlighting the challenges and benefits of doing this exploratory work in research libraries. **NEXT STEPS.** Academic research libraries are poised to play an important role in research and development around robust services for software curation. The next cohort of CLIR fellows are set to begin in fall 2018 and will likely shape and contribute substantially to an emergent research agenda.

Introduction

The evolving digital scholarly landscape has created new opportunities for academic research libraries and special collections to engage in the lifecycle of scholarly research output. From production to dissemination, librarians and archivists are increasingly participating in research data management. Activities can range from advising on best practices for file organization, to assisting with the deposit of research data sets into repositories. The growth in research data management services should not come as a complete surprise; nearly a decade has passed since the Association of College and Research Libraries (ACRL) recommended that academic research libraries must “recast their identities in relation to the changing modes of knowledge creation and dissemination” to remain vital amidst changing scholarly information environments (American Library Association, 2007).

Indeed, the transition to networked digital scholarship in turn resulted in new forms of scholarly research materials. In order to remain accessible and usable over time, digital media require curation – or active management and ongoing interventions throughout their lifecycle (Smith, 1999). For research libraries, managing these kinds of materials may require new competencies and workflows depending on scale and complexity. Examples of institutionally significant research output range from single database exports to multiple terabytes of observational data. Clarifying the various stakeholders and their particular needs around the creation, use, and reuse of software can play an important role in designing effective curation strategies for communities of practice.

In 2016, the Council on Library and Information Resources (CLIR) began offering a new postdoctoral fellowship research program in software curation, with placements at academic research libraries throughout the United States. This article describes the exploratory work undertaken by fellows at four different institutions from 2016-2018. We use our experiences and project work to highlight the role that research libraries can play in this domain, while also exposing potential pain points at intersecting boundaries of established practices and local needs.

Literature Review

Software as a Scholarly Research Object

The so-called data deluge has presented librarians and archivists with an opportunity to assist scholars in the active management of their digital content (Hey & Trefethen, 2008). Curating research output as institutional content is a relatively young, though growing phenomenon. Research data management (RDM) groups and services are increasingly common in research libraries, partially fueled by changes in federal funding grant application requirements to encourage data management planning. In fact, according to a recent content analysis of academic library websites, 185 libraries are now offering RDM services (Yoon & Schultz, 2017).

While research libraries are poised to assist researchers with organizing and managing their scholarly output, software presents challenges for institutions. Its dual nature as both a research object and a producer of research objects introduces conceptual complexities for the scholarly record across a number of domains. At the same time, research libraries have not yet integrated software into the wider digital scholarship ecosystem as a research object to be preserved and accessible for future use. Indeed, software is a relatively new area concern for data management groups in research libraries. Although there are no established practices currently, emerging community-driven efforts such as the PresQT Project and the Software Preservation Network reflect the growing acknowledgement that software plays an essential role in the research data lifecycle.

Our approach to software curation aligns with recent work on promoting scholarly research data as a “first class research object”, meaning a product validated, preserved, cited, and credited in a similar fashion as scholarly publications (De Roure et al., 2011; Belhajjame et al., 2012). In part, software’s prior omission may be attributable to the difficulties associated with modeling software in research workflows. Approaching software as a research object for reuse and reproducibility means acknowledging its dynamic state and interdependent nature, always in flux. As such, identifying persistent, significant characteristics about software– a cornerstone digital curation strategy – turns out to be a complex endeavor. Transforming software into a well-formed scholarly research object necessitates navigating murky waters where functions, definitions, and understandings may face translation challenges. Below we explore each of these areas in greater depth, highlighting potential issues and illustrating ways forward.

Curating Software in Research Libraries

What does it mean to curate and preserve software in academic research library settings? Exploring the types of software found in research library settings can help breakdown complexities in the curation space. Understanding the multiple stakeholders, uses, functions, and even locations of software can help to contextualize curation activities. Examples of software in these settings can run the gamut, from proprietary licensed materials to actively updated binaries on campus websites. Faculty members and students may create software as part of coursework or use/reuse software in the process of conducting new research. Software can also exist as part of manuscript collections donated by researchers or faculty affiliated with an institution. Software-driven artworks and websites are increasingly commonplace, with digital lab spaces and collaborative projects pushing disciplinary boundaries and practices. All of these scenarios likely involve software playing a significant role in the institutional scholarly record.

As an exemplar case, consider the research practices and output of faculty member Alice, who produces research tools and methodologies for data analysis. Documenting the components used and created by Alice for a particular research project might include the following:

- Primary data collected and used in analysis
- Secondary data collected and used in analysis
- Primary data output result(s) produced by analysis
- Secondary data output result(s) produced by analysis
- Software program(s) for computing published results
- Dependencies for software program(s) for replicating published results
- Published journal article

These components represent (at least) two particular instantiations of scholarly research workflows and output. First, obtaining verification of statistical results from primary data collected and analyzed occurs through replicating the conditions of the original analysis. Second, the statistical approach executed by the software program can analyze a new, “secondary” set of data and produce a secondary data output. This example demonstrates how software can simultaneously serve as both an outcome to be preserved and as a methodological means to an (new) end.

A common approach used in curation and preservation of digital objects involves characterizing the significant properties or “essence” of objects – as a means of identifying potentially meaningful aspects for different scholarly communities (Heslop, Davis, & Wilson, 2002; Hedstrom & Lee, 2002; Giaretta et al., 2009). Selected preservation strategies can be used for different purposes that communities deem valuable. Applying this logic to software, however, elicits a number of conceptual difficulties. What are the essential components of software that render it an object of scholarly significance? Is software *only* software in an executable state? Defining the boundaries of software - what constitutes its essence as a scholarly object to be preserved - is difficult in part because software has multiple definitions and diverse conceptions across disciplines. For example, we can think of software as an artifact (Kirschenbaum, 2013), as documentation of historical evidence (Bearman, 1989), or as “a collection of computer programs, programs, procedures and documentation that perform some task on a computer system” (Matthews et al., 2008). How we think about software today may be quite different then how we think about it tomorrow.

Another conceptual challenge emerges when the notion of curating software is introduced, a set of practices not yet defined in the literature. The term “curation” and its preceding choice of nouns carries a variety of meanings depending on the audience (Palmer et al., 2013). Some definitions emphasize the active, ongoing management of data to ensure future use (Lord et al., 2004; Cragin et al., 2008) while other definitions highlight the preservation of digital content (Beagrie, 2004; Yakel, 2004). The extent to which there are discrepancies in prescribed curation activities for each definition is not clear. Moreover, which set of practices are best suited for software, a complex digital object that exists dually as both an active producer of data and an artifact of digital information?

At the same time, motivations for software preservation can be far-reaching and heterogeneous. Historically, archivists have made the case that software provides documentary evidence of institutional histories and scientific research processes (Hess, 1985; Bearman, 1985). Academic researchers studying computer game preservation note the ability of computer game software to demonstrate new forms of immersive storytelling (McDonaugh et al., 2010; Kaltman et al., 2014). Scientific communities concerned with the validity of research output recommend preserving software to assist in the integrity and transparency of research endeavors (Allen & Schmidt, 2015). Identifying and mapping local needs to anticipated software functionalities is a critical aspect of software curation work.

Shifting expectations around the management and sharing of scholarly products like software and data present challenges for both researchers and research service providers. For example, consider the heterogeneous data sharing policies enforced by academic publishers and research funding bodies (Kriesberg et al., 2017; Vasilevsky et al., 2017). Policies pertaining to research-related code have historically lagged behind those related to data (Stodden et al., 2013), but as expectations related to the management and sharing of data evolved, so too have those related to software and code. For example, the Wellcome Trust now requires that researchers make available any original software that is required to view datasets or replicate analyses (Wellcome Foundation, 2017)

The highly iterative and collaborative nature of research software development means its production may not easily fit within existing academic incentive structures, which tend to emphasize the publication of novel and positive results (Nosek et al., 2012). In the scholarly record, references to software are inconsistent and important information about versions, parameters, and runtime environments is often missing entirely (Howison & Bullard, 2016). This not only poses a significant challenge for efforts aimed at ensuring computational reproducibility, but also makes tracking authorship, usage, and distribution extremely difficult. In 2016, the Force11 working group developed a set of software citation principles in order to encourage consistent policies for software citation across disciplines and venues; however, it has not yet seen widespread implementation (Smith et al., 2016).

There have been numerous calls urging researchers to thoroughly describe and share their software (e.g., Ince et al., 2012; Joppa et al., 2013; Morin et al., 2012). Yet the lack of formal mandates or requirements for doing so make such calls ineffectual. Complicating matters further, the expertise and infrastructure available to researchers is currently quite heterogeneous. Because many researchers lack formal training in software development practices, a recent set of guidelines advocates for “good enough” practices (Wilson et al., 2017). While packaging and containerization platforms such as ReproZip and Docker enable the tracking, bundling, and sharing of software libraries and dependencies, managing output means confronting the same curation difficulties (Emsley & De Roure, 2017). A final point is that through their integration with Github, services like Figshare and Zenodo allow researchers to deposit and receive a persistent identifier for their software. However, these services preserve a snapshot view of a dynamic object (software) at a single moment in time.

Unlike publications and datasets, software is executable, highly iterative, and often inter-dependent (Matthews et al., 2009). It relies on multiple dynamic elements, including the build and execution environment; dependencies and integrated libraries; metadata and specifications;

and the structure of source code and individual components that support functionality. All of these components are necessary during the software lifecycle for execution (Rios, 2016). At the same time, the essential components for reuse may differ according to the community of interest (Chue Hong, 2014). Factors that influence reuse of software include the quality of documentation and implementation details (Hucka & Graham, 2016). Additionally, having full access to the data used in research is crucial for ensuring reproducibility.

Software is potentially always evolving. Adding a new script can mean adding multiple dependencies such as new software libraries (Thain et al., 2015). Application software and underlying operating systems can also rapidly change. The occurrence of deprecated software and/or libraries is increasingly commonplace in the software development space. Migrating software successfully requires a great deal of effort and care. The challenge grows when the software is distributed or built in a collaborative environment (e.g., a complex university computing system).

The challenges of curating software are not just limited to simply providing access to, and preservation of, the "bits." Enabling adequate use and reuse of software involves technical interventions across the lifecycle, including specifying adequate metadata and capturing appropriate contextual information about both software and its original environment to facilitate different preservation strategies (e.g., execution, migration, emulation). The lack of applicable robust frameworks for preserving software as a complex digital object represents a significant challenge for sustainable access.

DESCRIPTION OF PROGRAM

Four institutions, Yale University Libraries, University of California Berkeley Libraries, California Digital Library, and The Massachusetts Institute of Technology Libraries, applied to host software curation fellows for a two-year period beginning in September 2016. Each institution designed the scope and range of research in conjunction with fellows' interests and background. For example, research projects at Yale and MIT focused on curating software within a special collections setting, while the California Digital Library and UC Berkeley investigated software curation in the context of the larger research process. These perspectives complimented each other and provided a useful base to anticipate curatorial interventions in the research software lifecycle.

Below we briefly describe the exploratory work fellows undertook at their site. Our goal is to demonstrate how software curation intersects with existing practices in contemporary academic research libraries. We also want to highlight some of the complexities of building out software-curation related services in libraries, in the hopes that we can extend our lessons learned to others seeking to develop similar services.

Yale University Library

The Library has begun a new program of work for systematically preserving software, in order to support long-term access and use of digital collections. A central goal of our work is to enable a representative interaction environment for all digital objects in Yale's collection. In particular, we are exploring the application and use of emulation tools and services to legacy collections on floppy disks and CD-ROMs. Our research asks the following question: can we provide emulation as a service to our library community?

In our current project, we use the bwFLA Emulation as a Service (EaaS) technology as part of a larger digital preservation workflow (see Figure 1). Our curation process begins with creating a disk image of legacy software, often residing on source media at risk of degradation. We ingest the image into our digital preservation system and use tools like DROID and

Siegfriend to identify file formats in an object. The EaaS framework communicates with our digital preservation system, taking the list of formats and using a Wikidata client to find the known software titles capable of reading those formats. EaaS then attempts to see if there is already a preconfigured software environment for the user.

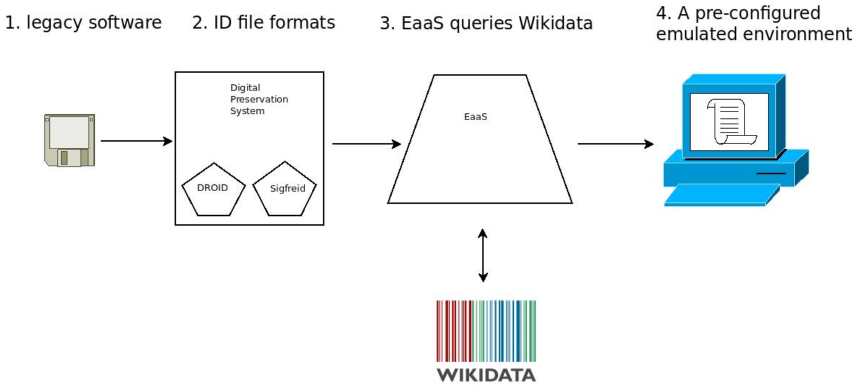

Figure 1. *Workflow for Emulation Service*

Another project that compliments ongoing EaaS work is The Wikidata for Digital Preservation Portal. The application supports contributed structured data to Wikidata related to file formats, software, emulated computational environments, and computer hardware. It also provides automated searching across the platform on similar topics. Inspired by Wikigenomes, an existing application created by the Su Lab (Putnam et al., 2017), the portal can be used on any supported browser. Contributed structured data will then be added to Wikidata through authenticated user accounts.

UC Berkeley Libraries

At UC Berkeley Library, our project focuses on developing frameworks to facilitate software sharing and preservation, and to encourage reproducibility and open science efforts. We also contribute to efforts aimed at ensuring the sustainability of research software, advocating for its treatment as a “first-class” research product. In spring 2017, we conducted an online survey consisting of 56 questions and addressing three main research questions. First, what are researchers doing with their code? Second, how do researchers share their code? Third, what do researchers value about their code? The total number of participants in our study was 330 and represented a wide variety of research disciplines. We recently completed our initial analysis and posted it as a preprint (see: AlNoamany and Borghi, 2018).

At UC Berkeley, the Research Data Management team will use our survey results to develop services for encouraging and sharing research software. We are pursuing three objectives in our ongoing collaboration.

1. Build consensus on key issues facing researchers and shape strategies for software preservation, including software citation.
2. Provide valuable information on researchers needs in order to adopt an agile approach that considers research software as a “first-class" research entity, with significant characteristics at multiple levels.
3. Draw conclusions to help service providers shape strategies for managing, preserving, and citing research software.

We have also been working closely on a project together to develop curricula for teaching researchers how to manage their research data and software. We adopted and modified an existing framework for research workflow (see Figure 2; adapted from Kubilius, 2014) and plan to use this as a roadmap, generating a rubric for researchers managing their data and software.

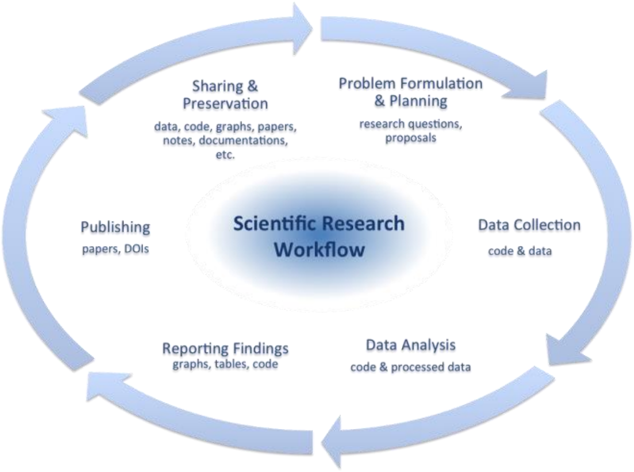
Sharing & Preservation
data, code, graphs, papers, notes, documentations, etc.
Problem Formulation & Planning
research questions, proposals
Publishing
papers, DOIs
Scientific Research Workflow
Data Collection
code & data
Reporting Findings
graphs, tables, code
Data Analysis
code & processed data

Figure 2. *Scientific Research Workflow*

In collaboration with the Software Carpentry community, we are providing a series of hands-on software and data management workshops for librarians to develop their skills. These workshops will introduce trainings focusing explicitly on the needs and requirements of library professionals. Our perspective is that empowering librarians with effective and reproducible computational skills will positively influence and serve the needs of research scientists and the larger research community. Initially, we plan to focus on foundational software skills in weekly/biweekly three-hour sessions. With these efforts, we hope to contribute positively toward empowering the library and to build a community with software and data management skills.

California Digital Library

At California Digital Library, our primary focus is on understanding software and computer code as a research product akin to, though distinct from, data and traditional scholarly publications (e.g., journal articles). Through our collaboration with UC Berkeley, we are developing a greater understanding of how researchers use, share, and value their computational tools. This, in turn, has informed how we frame such tools in the context of research and outreach projects related to research data management.

One of our central efforts has focused on surveying data management practices in magnetic resonance imaging (MRI) research settings. MRI is currently one of the most popular techniques for studying the structure and function of the brain. At present, the collection, processing, and analysis of MRI data requires the use of a wide range of computational tools, including community-developed software packages. Recent discovery of errors in these packages that have potentially wide-reaching effects on measures of statistical significance (Eickhoff et al., 2016; Eklund et al., 2016) has contributed to concerns about the rigor and reproducibility of cognitive neuroscience and related research areas. In response, MRI researchers have begun to converge on a set of best practices for managing and sharing their data, software, and other research products (Nichols et al., 2017). However, as of this writing, information about the extent to which these recommendations have actually been applied remains mostly anecdotal.

In collaboration with the Carnegie Mellon University Library, we have designed and distributed a survey that examines how and why MRI researchers manage their data and code throughout the course of a research project. Drawing from maturity-based models for assessing data management-related activities (Crowston & Qin, 2011) our survey addresses topics such as planning, documentation, organization, preservation, and sharing using language and terminology specific to MRI researchers. The goals of this project are to provide the digital curation community with a well-characterized use case for how researchers in a computationally complex research area manage their data and code and provide the MRI research community with expertise and empirical information that could inform future best practice recommendations. Our survey instrument may also serve as the basis for future research investigating the practices and perceptions of cognate research areas such as psychology.

Our second project focuses on building a data management guide for researchers. To help researchers navigate ever-changing expectations, we are developing a suite of tools for researchers to evaluate how they manage their research materials throughout their workflow. At present, these tools include a rubric designed to allow researchers and curation specialists to assess current practices and a series of one-page guides designed to help researchers advance their practices as desired or required. Similar to the research data lifecycle (Carlson, 2014), these tools frame management and sharing as continuous and iterative processes, where practices established at one stage of a research project are informed by those at earlier and subsequent stages. Because different research communities have different practices and perceptions regarding how their data should be managed (Akers and Doty, 2013), we are also working to ensure that both the rubric and the guides can be adapted to suit local needs and services.

Massachusetts Institute of Technology

The goal of our research is to identify, understand and describe baseline characteristics about software creation, use, and reuse, grounded in use cases found across MIT. Inspired by the Data Curation Profiles Toolkit to “tell the story of data”, we are developing a set of curation strategies that can be used by research libraries for software collection, curation, and preservation. Importantly, we wanted to adopt an agile research approach that considered software as an artifact as well as (simply) as an outcome to be preserved and made accessible. Curation in this sense might seek to answer ontological questions about software with significant characteristics at different levels of representation. We also wanted our approach to address essential curatorial activities as “interventions” to help ensure digital files remain accessible and usable (Smith, 2000). A central motivation for our work was to address noted gaps in existing curation approaches for complex digital objects like software.

During our first few months, we devised a set of exploratory research questions and began to identify and develop research approaches. One of our early exercises involved mapping out different scenarios for software use at MIT. We created multiple scenarios that grouped together entities with possible activities, purposes, functions, and uses (see Figure 3). Each scenario linked together possible activities with different potential purposes. This exercise proved fruitful for articulating the range of pathways for software use. Identifying the players in the ecosystem helped to characterize and produce a baseline understanding of the universe.

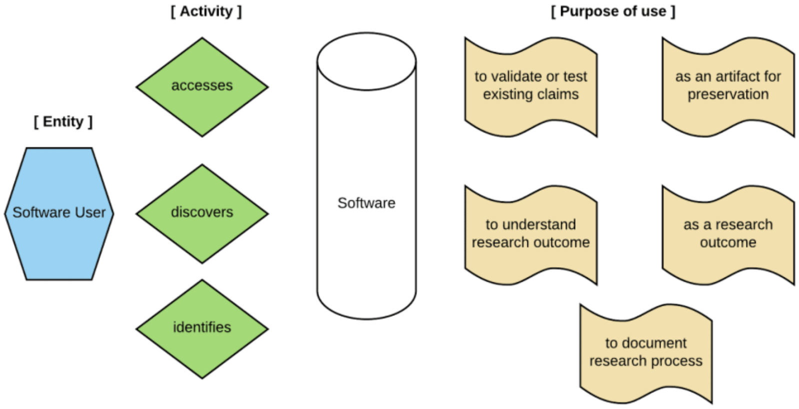
[Activity]
[Purpose of use]
[Entity]
Software User
accesses
discovers
identifies
Software
to validate or test existing claims
as an artifact for preservation
to understand research outcome
as a research outcome
to document research process

***Figure 3**. Software Scenarios*

As an institution with a rich history of computing and technological innovation, MIT has multiple examples of legacy and active software across campus in a variety of formats, locations, and conditions. The next step in our process was to identify and describe representative types of software and envision potential researcher scenarios. This work began with a literature review to surface possibilities, and concluded with an environmental scan across MIT to locate potential use cases for further exploration.

Below is a brief overview of our research process, detailing research questions and our corresponding research activities (see Table 1). In our remaining work, we expect to finalize the development of templates for devising Software Curation Profiles - a lightweight tool to provide guidance for library curators gathering information from software creators/owners.

Research questions	**Activities**
RQ1: How can we characterize software as a well-formed research object in an academic research library setting?	• Review literature on software collecting and use in research library settings • Identify and document representative types of software created, used, and reused at MIT • Characterize the state of software at MIT
RQ2: How do members of the research communities in disciplines represented at MIT use software in their activities? How might these distinctions inform strategies for curating software?	• Review literature on how researchers create, use, and reuse software • Identify and document representative uses for software throughout MIT campus • Characterize software creation, use, and reuse in research contexts by MIT affiliates.

Table 1. *Software curation research questions and activities*

NEXT STEPS

Academic research libraries have begun to institute research data management programs to provide assistance and guidance to researchers. While curation practices for data have started to become standardized, software represents new opportunities and challenges. Curating software introduces a number of open research questions across social, technical, and conceptual boundaries. Exploratory, in-depth study and continued research are required to address software needs in different domains.

As an organization with deep ties to research libraries and special collections, CLIR is uniquely qualified to identify emergent needs and develop new programs and services to address those needs. Taken as a whole, our project work over the last few years demonstrates the extent to which software curation intersects with emerging needs and active practices of research libraries. At the same time, outcomes from our assorted projects also reflect how difficult it can be to build out sustainable software curation services across institutions. The challenges and possibilities of curating software presents interesting and novel opportunities for research libraries, particularly for those willing (and able) to experiment and innovate. Working in an emerging research space provided us with unique opportunity to help formulate what it means to curate software in research library contexts.

CONCLUSION

The next cohort of CLIR software curation fellows will begin their tenure in the fall of 2018, presumably addressing similar areas of concern within different domains. In recounting our experiences, we hope to demonstrate the value of this kind of applied research for postdoctoral work. We also want to emphasize the importance of approaching software curation and preservation from cross-disciplinary perspectives, through broad coalitions of information professionals, domain experts, and communities of practice. Our work has benefited from the leadership provided by community-driven organizations such as the Software Sustainability Institute and the Software Preservation Network. Despite their newness, these groups have provided a crucial gathering space for sharing resources, constructing recommendations for best practices, and facilitating educational opportunities for the growing community of practice around software curation. We hope to see more community-driven initiatives coming out of alliances between software enthusiasts and information professionals. Curating software for long-term access and preservation is a wicked problem, requiring all hands on deck.

REFERENCES

Akers, K. G. & Doty, J. (2013). Disciplinary differences in faculty research data management practices and perspectives. *International Journal of Digital Curation*, *8*(2), 5–26. http://doi.org/10.2218/ijdc.v8i2.263

Allen, A. & Schmidt, J. (2015). Looking before leaping: creating a software registry. Retrieved January 8, 2018, from https://arxiv.org/pdf/1407.5378.pdf 2015

AlNoamany, Y. & Borghi, J. (2018). Towards computational reproducibility: researcher perspectives on the use and sharing of software. *PeerJ Preprints,* https://doi.org/10.7287/peerj.preprints.26727v1

American Library Association. (2007). The changing roles of academic and research libraries. Association for College and Research Libraries. Retrieved January 21, 2018, from http://www.ala.org/acrl/issues/value/changingroles

Beagrie, N. (2004). Digital curation centre. *Learned Publishing 17* (1).

Bearman, D. (1985). Collecting software: A new challenge for archives and museums. *Archives & Museum Informatics 2.*

Bearman, D. (1989). The case for software as documentation. *Iassist Quarterly*. Retrieved January 12, 2018, from http://www.iassistdata.org/sites/default/files/iqvol131bearman.pdf

Carlson, J. (2014). The use of lifecycle models in developing and supporting data services. In J. M. Ray (Ed.), *Research data management: Practical Strategies for information professionals (*pp. 63–86). West Lafayette, Indiana: Purdue University Press.

Chue Hong, N. (2014). Minimal information for reusable scientific software. *Proceedings of the 2nd Workshop on Working towards Sustainable Scientific Software: Practice and Experience*. doi:10.6084/m9.figshare.1112528.

Cragin, M., Palmer, C.L., Carlson, J.R. & Witt, M. (2010). Data sharing, small science, and institutional repositories. *e-Science: Past, Present, and Future ii 368*(1926).

Crowston, K., & Qin, J. (2011). A capability maturity model for scientific data management: Evidence from the literature. *Proceedings of the American Society for Information Science and Technology*, *48*(1), 1–9. http://doi.org/10.1002/meet.2011.14504801036

Di Cosmo, R. & Zacchiroli, S. (2017). Software heritage: Why and how to preserve software source code. *Proceedings of the Fourteenth International Conference on Digital Preservation (iPRES), Kyoto, Japan, September 2017.*

Eickhoff, S. B., Laird, A. R., Fox, P. M., Lancaster, J. L., & Fox, P. T. (2017). Implementation errors in the GingerALE Software: Description and recommendations. *Human Brain Mapping*, *38*(1), 7–11. http://doi.org/10.1002/hbm.23342

Eklund, A., Nichols, T. E., & Knutsson, H. (2016). Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates. *Proceedings of the National Academy of Sciences*, *113*(28), 7900–7905. http://doi.org/10.1073/pnas.1602413113

Giaretta, D., Matthews, B., Bicarregui, J., Lambert, S., Guercio, M., Michetti, G., & Sawyer, D. (2009). Significant properties, authenticity, provenance, representation information and OAIS information. *iPRES 2009: The Sixth International Conference on Preservation of Digital Objects (iPRES)*, October 5-6, 2009, San Francisco, CA.

Hedstrom, M., & Lee, C.A. (2002). Significant properties of digital objects: Definitions, applications, implications. *Proceedings of the DLM-Forum.* Retrieved January 12, 2018, from https://ils.unc.edu/callee/sigprops_dlm2002.pdf

Heslop, H., Davis, S., & Wilson, A. (2002). An approach to the preservation of digital records. National Archives of Australia. http://pandora.nla.gov.au/tep/49636

Hess, J.K., Samuels, H.W., & Simmons, B.T.(1985). Appraising the Records of Modern Science and Technology: A Guide," Cambridge, MA: Massachusetts Institute of Technology, 1985;

Hucka, M. & Graham, M.J. (2016). Software search is not a science, even among scientists. *Computing Resources Repository*. https://arxiv.org/abs/1605.02265

Howison, J., & Bullard, J. (2016). Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature. *Journal of the Association for Information Science and Technology*, *67*(9), 2137–2155. http://doi.org/10.1002/asi.23538

Ince, D. C., Hatton, L., & Graham-Cumming, J. (2012). The case for open computer programs. *Nature*, *482*, 485. http://doi.org/10.1038/nature10836

Joppa, L. N., McInerny, G., Harper, R., Salido, L., Takeda, K., O'Hara, K., … Emmott, S. (2013). Troubling trends in scientific software use. *Science*, *340*(6134), 814–815. http://doi.org/10.1126/science.1231535

Kaltman, E., Wardrip-Fruin, N., Lowood, H., & Caldwell, C. (2014). A unified approach to preserving cultural software objects and their development histories. Retrieved January 12, 2018, from https://escholarship.org/uc/item/0wg4w6b9

Katz, D.S., Niemeyer, K. E., Smith, A.M., Anderson, W.L., Boettiger, C. Hinsen, K., Hooft, R., Hucka, M., Lee, A., & Löffler, F. (2016). Software vs. data in the context of citation. Technical Report, *PeerJ Preprints*.

Kirschenbaum, M. (2013). An executable past: The case for a national software registry. *Preserving.exe: Toward a national strategy for preserving software*. Library of Congress. http://lcweb2.loc.gov/master/gdc/lcpubs/2013655114.pdf

Kriesberg, A., Huller, K., Punzalan, R., & Parr, C. (2017). An analysis of federal policy on public access to scientific research data. *Data Science Journal, 16*(27), 1-13. http://doi.org/10.5334/dsj-2017-027

Kubilius, J. (2014). A framework for streamlining research workflow in neuroscience and psychology. *Frontiers in Neuroinformatics 7*. https://doi.org/10.3389/fninf.2013.00052

Lord, P., Macdonald, A., Lyon, L., & Giaretta, D. (2004). From deluge to data curation. *Escience All Hands Meeting*, August 31-September 3, 2004, Nottingham, UK. Retrieved January 10, 2018, from http://www.ukoln.ac.uk/ukoln/staff/e.j.lyon/150.pdf

Lowood, H. (2013). The lures of software preservation. *Preserving.exe: Toward a national strategy for preserving software*. Library of Congress. http://lcweb2.loc.gov/master/gdc/lcpubs/2013655114.pdf

Matthews, B., McIlwrath, B., Giaretta, D., & Conway, E., (2008). The significant properties of software: A study. Retrieved January 12, 2018, from http://purl.org/net/epubs/work/65878

Matthews, B., Shaon, A., Bicarregui, J., & Jones, C. (2010). A framework for software preservation. *International Journal of Digital Curation 1*, 91–105.

McDonaugh, J., Olendorf, R., Kirschenbaum, M., Kraus, K., Reside, D., Donahue, R., Phelps, A., Egert, C., Lowood, H., & Rojo, S. (2010). *Preserving Virtual Worlds: Final Report*. Retrieved January 12, 2018, from https://www.ideals.illinois.edu/bitstream/handle/2142/17097/PVW.FinalReport.pdf

Meyerson, J., Vowell, Z., Hagenmaier, W., Leventhal, A., Rios, F., Russey Roke, E., & Walsh, T. (2017). The Software Preservation Network (SPN): A community effort to ensure long term access to digital cultural heritage. *D-Lib Magazine* 23 (5/6). https://doi.org/10.1045/may2017-meyerson

Morin, A., Urban, J., Adams, P. D., Foster, I., Sali, A., Baker, D., & Sliz, P. (2012). Shining light into black boxes. *Science, 336*(6078), 159-160. http://doi.org/10.1126/science.1218263

Nosek, B. A., Spies, J. R., & Motyl, M. (2012). Scientific utopia: II. Restructuring incentives and practices to promote truth over publishability. *Perspectives on Psychological Science, 7*(6), 615–631. http://doi.org/10.1177/1745691612459058

Palmer, C.L., Weber, N., Munoz, T., & Renear, A., (2013). Foundations of data curation work. *Archive Journal 3*. Retrieved January 12, 2018, from http://www.archivejournal.net/essays/foundations-of-data-curation-the-pedagogy-and-practice-of-purposeful-work-with-research-data/

Putman, T. E., Lelong, S., Burgstaller-Muehlbacher, S., Waagmeester, A., Diesh, C., Dunn, N., Munoz-Torres, M., Stupp, G.S., Wu, C., Su, A.I., & Good, B. M. (2017). WikiGenomes: an open web application for community consumption and curation of gene annotation data in Wikidata. *Database*, https://doi.org/10.1093/database/bax025

Rios, F. (2016). The pathways of research software preservation: An educational and planning resource for service development. *D-Lib Magazine 22*(7/8).

Smith, A. (2002). Authenticity in perspective. *Authenticity in a Digital Environment,* Council on Library and Information Resources. Retrieved January 03, 2018, from https://www.clir.org/pubs/reports/pub92/smith/

Smith, A. M., Katz, D. S., & Niemeyer, K. E. (2016). Software citation principles. *PeerJ Computer Science, 2*, e86. http://doi.org/10.7717/peerj-cs.86

Stodden, V., Guo, P., & Ma, Z. (2013). Toward reproducible computational research: An empirical analysis of data and code policy adoption by journals. *PLOS ONE*, *8*(6), 2–9. http://doi.org/10.1371/journal.pone.0067111

Thain, D., Ivie, P., & Meng, H. (2015). Techniques for preserving scientific software executions: Preserve the mess or encourage cleanliness? *iPRES 2015: 12th International Conference on Digital Preservation,* Nov 2-6, 2015, Chapel Hill, North Carolina.

Tenopir, C., Sandusky, R.J., Allard, S., & Birch, B. (2014). Research data management services in academic research libraries and perceptions of librarians. *Library & Information Science Research 36*(2), 84-90. https://doi.org/10.1016/j.lisr.2013.11.003

Tenopir, C., Dalton, E.D., Allard, S., Frame, M., Pjesivac, I., Birch, B., Pollock, D., & Dorsett, K. (2015). Changes in data sharing and data reuse practices and perceptions among scientists worldwide. *PLOS ONE, 10*(8). https://doi.org/10.1371/journal.pone.0134826

Vasilevsky, N. A., Minnier, J., Haendel, M. A., & Champieux, R. E. (2017). Reproducible and reusable research: are journal data sharing policies meeting the mark? *PeerJ, 5*, e3208. http://doi.org/10.7717/peerj.3208

Wellcome Foundation (2017, July 10). *Policy on data, software and materials management and sharing*. Retrieved from https://wellcome.ac.uk/funding/managing-grant/policy-data-software-materials-management-and-sharing

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., & Teal, T. K. (2017). Good enough practices in scientific computing. *PLOS Computational Biology, 13*(6), e1005510. https://doi.org/10.1371/journal.pcbi.1005510

Witt, M., Carlson, J., Brandt, D.S., & Cragin, M. (2009). Constructing data curation profiles. *International Journal of Digital Curation 4*(3).

Yakel, E. (2007). Digital curation. *OCLC Systems & Services: International Digital Library Perspectives 23*(4), 335-340. https://doi.org/10.1108/10650750710831466

Yoon, A., & Schultz, T. (2017). Research data management services in academic libraries in the US: A content analysis of libraries' websites. *College & Research Libraries 78* (7). 10.5860/crl.78.7.920.