The challenge and promise of software citation for credit, identification, discovery, and reuse

Kyle E. Niemeyer

School of Mechanical, Industrial, and Manufacturing Engineering,
Oregon State University
kyle.niemeyer@oregonstate.edu

Arfon M. Smith

GitHub Inc.
arfon@github.com

Daniel S. Katz

National Center for Supercomputing Applications &
Electrical and Computer Engineering Department & School of Information Sciences,
University of Illinois at Urbana–Champaign
d.katz@ieee.org

1 Introduction

Modern science and engineering research depends on software. A 2009 survey of scientists found that 91% consider software important or very important to their research [Hannay et al., 2009]. The scientific community uses citation to acknowledge traditional research results published in archival journals and conferences, but no such accepted standard exists to credit the considerable efforts that go into software—and most research software is not cited [Pan et al., 2015].

One method to increase the amount of software developed, shared, and credited is to treat a software release as a publication. There is good evidence that academics respond to incentives, including interview and survey data saying that increased citation would drive increased software development and sharing [Howison and Herbsleb, 2011; Huang et al., 2013]. Furthermore, evidence indicates that research activity increases when outputs can be formally counted [McNaught, 2015]. Finally, evidence of success in direct citations to datasets [Belter, 2014]—as opposed to indirect citations through publications—suggests similar positive benefits for software citation.

In addition to the above benefits, since research results depend on the specific software used (e.g., version), proper citation—and the associated preservation—is necessary to ensure reproducibility [Sufi et al., 2014]. Provenance of research results and data requires, among other things, a record of the software used to generate or process that data [Sandve et al., 2013; Wilson et al., 2014]. Potential errors in software or variations due to environment [Morin et al., 2012; Soergel, 2015] further warrant the citation of the specific software used.

Different communities follow widely varying practices for citing software, with guidelines ranging from citations of an associated paper or the software itself via DOI [American Astronomical Society, 2016] to no policy at all, with many ad hoc practices in between [Howison and Bullard, 2015]. Furthermore, some research communities have not yet adopted open-source mentalities regarding research software. Questions also remain about the role of curating/reviewing software—if software will be cited in the same manner as a publication, is quality assessment needed (e.g., peer code review)? If so, how, when, and by whom?

The main challenge of software citation is the lack of a way to uniquely identify released/ published software, so those who use it can cite it. We also need a process to curate and review software. As a first step towards sustainable, reusable, and attributable software, efforts are underway to establish citation practices for software used in research.

2 Related Work

Multiple organizations, including the WSSSPE Software Credit working group [Katz et al., 2014, 2016a,b] and FORCE11 Software Citation Working Group [Smith et al., 2016], are working to standardize software citation practices. Some of the observations and recommendations made here come from their efforts, which follow similar work by DataCite and the FORCE11 Data Citation Synthesis Group [2014] to standardize research data citation practices. More detail on these efforts, as well as broader related work by the community aimed at both citation and credit for software, can be found in the declaration of Software Citation Principles [Smith et al., 2016].

Even with an adopted, standard method for software citation, indexers (e.g., Web of Science, Scopus, Google Scholar) currently lack support for indexing such citations. The astrophysics community represents a possible exception, where the community-run service NASA ADS [Accomazzi et al., 2016] also carries out indexing of citations. The ADS already indexes references to software listed in the Astrophysics Source Code Library [2016], and has made a commitment to expand upon this functionality following an AAS/ GitHub-sponsored meeting [Norén, 2015].

Finally, peer review of software distinctly lacks a standard practice. Some journals that accept both "software" submissions and general research papers can conflate review of the submitted software with a review of research output **produced by** the software. Software-only journals (e.g., *Journal of Open Research Software*, *SoftwareX*) and the rOpenSci community represent exceptions to this, with documented review processes. For example, rOpenSci has a process [Ross et al., 2016] used to determine whether a package can become part of the rOpenSci collection.

3 Challenges and Research Directions

Table 1 summarizes key challenges and research directions for software citation, along with possible solutions/ methods.

Table 1: Key research challenges and possible solutions/methods

Key research challenge	**Possible solutions/methods**
Identify necessary metadata associated with software for citation	We suggest metadata below, and the CodeMeta project [Boettiger et al., 2016] is working to determine minimal metadata.
Standardize proper formats for citing software in publications	The FORCE11 Software Citation Working Group is defining, and gaining community acceptance for, software citation principles, after which a follow-on group will begin implementation efforts.
Establish mechanisms for software to cite other software (i.e., dependencies)	Software publication as software papers allows this. For software that is directly published, this is an open challenge.
Develop infrastructure to support indexing of software citations within the existing publication citation ecosystem	The FORCE11 Software Citation Implementation Working Group will also work on this, in collaboration with publishers and indexers.
Determine standard practices for peer review of software	Professional societies and science communities need to determine how this will happen.
Increase cultural acceptance of the concept of software as a digital product	Acceptance will happen over time; unclear how to accelerate this process.

For software to be cited, we recommend that metadata include software name, primary authors/contributors (name and ORCID), DOI or other unique and persistent identifier, and location where the software has been published/archived. Typically, DOI (or a similar unique identifier) will provide both identification and location, but if not then metadata should include a way to locate the software (e.g., URL). This information should be provided in a `CITATION` file, potentially in JSON or XML format (and with an appropriate metadata schema, e.g., DOAP) to allow automatic processing. Citations of software in publications should minimally include software name, primary authors, and DOI or location where the software was published/archived; however, individual citation formats will vary based on the particular style of journals, conferences, or professional societies. Existing services such as Libraries.io [Nesbitt and Pompilio, 2016] and Depsy [Priem and Piwowar, 2015] automate software dependency tracking; these could be harnessed to produce citation networks. Technical solutions to these challenges exist—community acceptance is instead needed.

In arguing the need for software citation, we simultaneously introduce the question of **when** software should be cited. Although there is no clear answer to this question, ensuring reproducibility of research results requires citation of software if used directly and important to research results. In other words, if using different software could produce different data or results, then the software used should be cited.

Research and development efforts are needed to solve the remaining challenges. These include: how can we cite closed-source/commercial software—can the above information be provided, even if the software itself is not publicly preserved? Software citations require indexing to create a citation network akin to publications to carry weight for academic credit and reputation, so how can index services be encouraged to fully index software? How can publications indicate direct use of software for research in citations, where results would not be possible without efforts of software authors—should such a citation be "weighted" higher than others? The American Astronomical Society [2016] suggests a new "Software" section below the acknowledgements; other approaches include machine-readable supplementary data [Katz and Smith, 2015].

Finally, open questions remain on whether citable software should go through peer review and, if so, how can this be implemented? Should citable software itself follow the arXiv preprint model where releases are made available for users and the community to judge, or, alternatively, the software paper model where "advertising" papers undergo peer review in a relevant community?

Acknowledgments

Some work by K. E. Niemeyer was supported by NSF grant ACI-1535065. Work by D. S. Katz was supported by the National Science Foundation (NSF) while working at the Foundation; any opinion, finding, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF. The authors wish to thank the members of the SciSIP community and mailing list, in particular Chris Belter, James Howison, and Joshua Rosenbloom, for a useful discussion on incentives for software publications and how their impact has been and could further be measured. In addition, the authors acknowledge members of the FORCE11 Software Citation Working Group, whose views contributed to this document.

References

Alberto Accomazzi, Michael J Kurtz, and others. 2016. The SAO/NASA Astrophysics Data System. http://adswww.harvard.edu/. (2016). Accessed: 2016-04-07.

American Astronomical Society. 2016. Policy Statement on Software. http://journals.aas.org/policy/software.html. (Jan. 2016). Accessed: 2016-01-12.

Astrophysics Source Code Library. 2016. http://ascl.net. (2016). Accessed: 2016-04-07.

Christopher W. Belter. 2014. Measuring the Value of Research Data: A Citation Analysis of Oceanographic Data Sets. *PLoS ONE* 9, 3 (03 2014), 1–9. DOI:http://dx.doi.org/10.1371/journal.pone.0092590

Carl Boettiger and others. 2016. CodeMeta. http://codemeta.github.io/. (2016). Accessed: 2016-04-10.

FORCE11 Data Citation Synthesis Group. 2014. Joint Declaration of Data Citation Principles. https://www.force11.org/group/joint-declaration-data-citation-principles-final. (2014). Maryann Martone, Editor. Accessed: 2015-10-28.

Jo Erskine Hannay, Hans Petter Langtangen, and others. 2009. How Do Scientists Develop and Use Scientific Software?. In *Proc. 2009 ICSE Workshop on Soft. Eng. for Computational Sci. and Eng. (SECSE)*. IEEE, Vancouver, BC, 1–8. DOI:http://dx.doi.org/10.1109/SECSE.2009.5069155

James Howison and Julia Bullard. 2015. Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature. *J. Assoc. for Information Science and Technology* (2015). DOI:http://dx.doi.org/10.1002/asi.23538 In press.

James Howison and James D. Herbsleb. 2011. Scientific Software Production: Incentives and Collaboration. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work (CSCW '11)*. ACM, New York, NY, USA, 513–522. DOI:http://dx.doi.org/10.1145/1958824.1958904

Xing Huang, Xianghua Ding, Charlotte P. Lee, Tun Lu, and Ning Gu. 2013. Meanings and Boundaries of Scientific Software Sharing. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. ACM, New York, NY, USA, 423–434. DOI:http://dx.doi.org/10.1145/2441776.2441825

Daniel S. Katz, Sou-Cheng T. Choi, and others. 2014. Summary of the First Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE1). *J. Open Research Software* 2, 1 (2014), e6. DOI:http://dx.doi.org/10.5334/jors.an

Daniel S. Katz, Sou-Cheng T. Choi, and others. 2016a. Report on the Second Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE2). *J. Open Research Software* 4, 1 (2016), e7. DOI:http://dx.doi.org/10.5334/jors.85

Daniel S. Katz, Sou-Cheng T. Choi, and others. 2016b. Report on the Third Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE3). (2016). arXiv:1602.02296 [cs.SE].

Daniel S. Katz and Arfon M Smith. 2015. Transitive Credit and JSON-LD. *J. Open Research Software* 3, 1 (2015), e7. DOI:http://dx.doi.org/10.5334/jors.by

Keith McNaught. 2015. The Changing Publication Practices in Academia: Inherent Uses and Issues in Open Access and Online Publishing and the Rise of Fraudulent Publications. *J. Electronic Publishing* 18 (2015). Issue 3. DOI:http://dx.doi.org/10.3998/3336451.0018.308

A Morin, J Urban, P D Adams, I Foster, A Sali, D Baker, and P Sliz. 2012. Shining Light into Black Boxes. *Science* 336, 6078 (2012), 159–160. DOI:http://dx.doi.org/10.1126/science.1218263

Andrew Nesbitt and Mauro Pompilio. 2016. Libraries.io. https://libraries.io. (2016). Accessed: 2016-01-17.

Laura Norén. 2015. Invitation to comment on a proposal for a cohesive research software citation-enabling platform. http://astronomy-software-index.github.io/2015-workshop/. (2015). Accessed: 2016-04-06.

Xuelian Pan, Erjia Yan, Qianqian Wang, and Weina Hua. 2015. Assessing the impact of software on science: A bootstrapped learning of software entities in full-text papers. *J. Informetrics* 9, 4 (2015), 860–871. DOI:http://dx.doi.org/10.1016/j.joi.2015.07.012

Jason Priem and Heather Piwowar. 2015. Depsy. http://depsy.org. (2015). Accessed: 2016-01-17.

Noam Ross, Carl Boettiger, and others. 2016. Onboarding at rOpenSci: A Year in Reviews. http://ropensci.org/blog/2016/03/28/software-review. (2016). Accessed: 2016-04-06.

Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, and Eivind Hovig. 2013. Ten Simple Rules for Reproducible Computational Research. *PLoS Comp Biol* 9, 10 (Oct. 2013), e1003285. DOI:http://dx.doi.org/10.1371/journal.pcbi.1003285

Arfon M Smith, Daniel S Katz, Kyle E Niemeyer, and FORCE11 Software Citation Working Group. 2016. Software Citation Principles. *PeerJ Preprints* 4 (2016), e2169v2. DOI:http://dx.doi.org/10.7287/peerj.preprints.2169v2

David A W Soergel. 2015. Rampant software errors may undermine scientific results [version 2; referees: 2 approved]. *F1000Research* 3 (2015), 303. DOI:http://dx.doi.org/10.12688/f1000research.5930.2

Shoaib Sufi, Neil P Chue Hong, and others. 2014. Software in Reproducible Research: Advice and Best Practice collected from experiences at the Collaborations Workshop. In *Proc. 1st ACM SIGPLAN Work. on Reproducible Research Methodologies and New Publication Models in Comp. Eng. (TRUST '14)*. ACM, Edinburgh, United Kingdom, 2:1–2:4. DOI:http://dx.doi.org/10.1145/2618137.2618140

Greg Wilson, D. A. Aruliah, and others. 2014. Best Practices for Scientific Computing. *PLoS Biol* 12, 1 (2014), e1001745. DOI:http://dx.doi.org/10.1371/journal.pbio.1001745