GOODLUCK ASOBENIE KANDONGA,
JINGDA DING, YIQING YUAN

Moving toward sustainable software citation practices to improve the quality of scientific research

Software is essential for scientific research and is applied in research methodology, data analysis, and knowledge dissemination. Scientists believe that software plays a crucial role in their research process, so it is necessary to recognize the contribution of software developers. This study aims to investigate the current situation of software citation in scientific research and explore possible solutions to improve its sustainability via full-text content analysis. There is a gap between the use and the citation of software in scientific publications, and the lack of information in software citations has been a source of both failure and improvement of software in scientific research. We found various deficiencies that hinder the durability of software citations and put forward some suggestions for their stability and development, such as designing software citation standards, promoting free open licensing of scientific software, and implementing strict peer review of software citations.

Keywords: *Software citation, software mention, software use, software sustainability, software reward*

Aknowledgment

We would like to extend our sincere thanks to Shanghai University and all those involved in making this article a masterpiece.

Author Information

Goodluck Asobenie Kandonga, Shanghai University
https://ischool.shu.edu.cn/
Jingda Ding, Shanghai University
https://ischool.shu.edu.cn/
Yiqing Yuan, Shanghai University
https://ischool.shu.edu.cn/

How to cite this article:

Kandonga, Goodluck Asobenie, Jingda Ding, Yiqing Yuan. "Moving toward sustainable software citation practices to improve the quality of scientific research". *Információs Társadalom* XXII, no. 4 (2022): 49–63.
https://dx.doi.org/10.22503/inftars.XXII.2022.4.4

1. Introduction

With the rapid development of digital scholarship, scientific research increasingly relies on various kinds of software or computer programs. However, the productivity and the quality of software for scientific research generally fall short of expectations, as the software becomes dated and unable to adapt to continually changing requirements. One of the main reasons for these failures is that the incentives and rewards for software developers are insufficient; in addition, most researchers have little understanding of the software citation process. Silvello (2018) explains that citations are the cornerstone of knowledge dissemination and the primary means of both assessing the quality of research and guiding investment. Science is increasingly becoming data intensive, and a large amount of data is often collected and analyzed by different kinds of software to solve scientific problems or discover intricate patterns. Therefore, the importance of software references in academic research is evident. Some software developers are motivated to facilitate the design, development, use, testing, and application of their patches by sharing the program code with the user community, but they rarely receive credit via citations in publications; for example, research software is rarely cited in the Clarivate Analytics Data Citation Index (Park and Wolfram 2019).

Traditional reference sources are generally included in bibliographies, but research software is often excluded, even though it is executable and shows creative work, because metadata standards have yet to be formed. In addition, unlike other sources included in bibliographies, software references remain informal; for instance, the name or URL of the research software may be mentioned in the text or in the acknowledgments but not cited in the references, which sometimes leads to different objects referring to the same research software. Therefore, there is a gap between the use and the citation of software in scientific publications. The developers sharing their software code may not receive the appropriate credit or acknowledgment from peer-reviewed or non-peer-reviewed works such as user manuals, technical reports, or software landing pages(Hwang et al.2017). Consequently, issues have occurred regarding the visibility of software in scientific research, which may hinder the motivation of software developers. Now, it is time to take measures to acknowledge the importance of research software and discuss sustainable software citation solutions to improve the quality of scientific research.

2. Literature review

Relevant Research on Software Citations

Today, software can help scientists process and analyze scientific data, predict and test research hypotheses, improve scientific research efficiency, promote science development, etc. Therefore, the use and the reuse of software are essential in contemporary scientific research and academic exchanges. The communities that recognize and embrace the diversity of knowledge production also acknowledge software as a legitimate contribution to research. Howison and Herbsleb (2013) found that academic credit is a powerful motivator for the production and improvement of scientific software. As with the chicken and egg issues in data references (Mooney and Newton 2012), there is an imbalance between the development and the sharing of scientific software. Its development is usually proprietary rather than open, which runs counter to the ideal of the "Publicity of Science" and jeopardizes the ability to validate and reuse software (Gambardella and Hall 2006; Ince et al. 2012). Some solutions have been proposed to improve the sustainability of software, such as code-sharing infrastructures and quantitative measures for quality (Goble 2014). Meta software can reduce the barriers to using research software because it does not need complicated installation and configuration. Quantitative measure indexes, such as the number of lines of program code, are easy to calculate. There is a lack of certainty in showing the core characteristics of research software since the factoring of code can reduce the number of lines and since small changes in the core algorithm would have a significant impact on the function. Most metrics in software citation management do not properly credit the insights of the software development community (Abbott et al. 2010; Nightingale and Marshall 2012; Sahel 2011).

The detailed information included in reference citations plays a crucial role in tracking and contributing to the development and implementation of ideas. Moreover, sufficient information about the software used can promote the sustainability of software development and the improvement of academic research quality (Heinle et al. 2017). Knepley et al. (2013) suggest that research software developers should take steps to recommend citation preferences for citing their code via readme files, license agreements, login pages, user manuals, and other documentation. Some software tools or programming languages allow users and developers to execute codes that automatically generate the citation information; for example, the statistical programming language R has a function that helps to generate, compile, and quote information about contributors and their roles in software development in a fixed format. In addition, software used in the research process needs to be citation format, so that it can be adapted to fit with new developments in scientific research (Borgman et al. 2012) because information within the software citation can provide other researchers access to and identification of the research software. Examples include access to the version of the software used and the application of configuration settings. Software citation practices also need to introduce relevant information to support the validation, replication, and implementation of the software. However, the verification and the replication of research software require being able not only to find references but also to acquire and license them (Ince et al. 2012). Thus, the academic community should make software citation an essential part of research.

Current Software Citation Practices

According to the American Astronomical Society (2016), scientific articles should include standardized software citations to give the author credibility and give readers access to the exact version of the software used. However, some code repositories (such as GitHub) provide different types of software citations by a given platform. Meanwhile, the existence of various kinds of research software problematizes the requirements for software citation content. Due to the complexity inherent in citations and the different software citation standards, “meta software” has emerged to support the availability of software information in scientific research, as it can help researchers capture the information they cited and document how the software was used in their work and the development of the software. It also offers new opportunities to measure the impact of the software used on other factors, such as tenure review and promotion.

Receiving credit for creating code is just one of the many issues relating to the sustainability and repeatability of software development. Creating recognition systems to obtain credit for creating code can increase the visibility of the various contributors and make the coding process more transparent if the developers have provided detailed information for citation in the software. To this end, a campaign has been launched to recognize software developers in the scientific publishing community,which has already led to the introduction of useful software development work in the research support community by examining their ability to develop robust software documents. The development of the “Software Award”has been independently recognized as a means to document the publication and citation of software (Bangerth et al. 2016).

Software citation is also essential to research models. If the information about the software used in a study is lost, it will not be easy to verify that study’s results. The software citation should include not only the modeling package but also the complete relevant information about the software was used. Reproducibility is one of the positive features in academic research, but it often ignores the software used in the scientific process. Currently, more attention is given to approaches to generating new knowledge in academic research than to citing software.

Adoption of software citation practices depends on developing a guide to scholarly communication standardsfor the scientific research community. Additionally, most research using software does not mention how to cite the software correctly. If the software website, citation file, or readme file with the source code specifies how to reference the software, authors can use this information to cite the software in their publications (Katz and Hong 2018). Although software citation practices are currently not standardized or widely implemented, the publication of software reference guidelines has become the basis for building community guidelines and improving tools and infrastructure for supporting citations.

The challenge of monitoring citation infrastructure through the research literature is extensive, and no single tool or method can provide a solution that solves every problem. Interest has been devoted to software citation practices in scientific research, but the question remains as to which entities will enforce the standards regarding such citations in academic publications. The infrastructure of the research process has changed over time, leading to the introduction and disappearance of software tools (Mayernik et al. 2017). Currently, there is no standardized mechanism for evaluating software citations. For software developers and users, assessing contributions and acknowledging sources remain challenging. For example, a software engineer's contribution to the test software is to ensure that updates do not destroy the code, but that contribution is invisible in all current measurements. It is necessary to take some measures to encourage researchers to share more and promote the development of software in the research process. Software that is integrated into the researcher's workflow can help to facilitate more access, interpretation, and evaluation of research results.

Proper software citation practices in academic publications can support the excellent standards and content of the software, which can lead to readers using the cited software. Large libraries and applications, and even some open-source developers, have to list appropriate citation methods on their websites or documents to ensure that the software citation in a scientific publication is complete and accurate.

3. Methods

Data

The data used in this study were extracted from *Web of Science*. On January 22, 2019, we obtained 544 papers published between 2015 and 2017 through the search term "software reference," limiting the type of publication to "articles."*Web of Science* divides the results into many research areas including agricultural science, biology, medicine and health science, computer science, education and educational research, social science, management, and physics; however, some of these areas were very similar (for example, biology and biology and life sciences). Thus, we categorized these areas into nine disciplines based on similarity (see Table 1). Due to the nature of manual coding, we could include only a few articles in the database. We randomly selected 271 (50%) papers that were representative of these nine disciplines to check whether the patterns obtained from them are sufficient to describe the characteristics of the software citations.

Grouped disciplines	Number of articles
Agricultural science	9
Biology	27
Medicine and health science	44
Computer science	12
Earth science	8
Education and educational research	116
Social science	12
Management	23
Physics	20

Table 1.Discipline classification(Own editing)

Coding Procedure

We applied content analysis as a research tool. Content analysis is an effective method for finding quantitative patterns from textual data (Krippendorff 2004). Previous research used this approach to explore communication patterns that often centered on newspaper collections, journal publications, and online content, such as disciplinary discourse patterns (O'Connor et al 2017; Sugimoto et al. 2017), political activity strategy (Semetko and Valkenburg 2000). In content analysis, coding is the key to all processes. There is a link between data collection and interpretation that enables researchers to systematically understand data with a set of guidelines (i.e., coding schemes).

The first step of coding is identifying the research objectives and creating a transparent coding scheme. We drafted a coding scheme by targeting the research objectives shown in the introduction. The coding items were prearranged, and the latest code—the so-called emergency code—may appear throughout the code (Saldana 2009). In the process of encoding the 271 papers that we randomly selected from *Web of Science*, we marked the emergency code with "*" and applied this coding scheme to the collected data set. Each sentence of each paper is represented as a code analysis unit; coded items and explanations are listed in Table 2. For a few sentences that had different codes per paper, we finally obtained the same coding after rereading and negotiation.

Coded items	Explanations
1. Software use in research 1.1 Y 1.2 N	This item refers to whether the article used at least one software program during the research process. If so, 1.1 was used to encode and continue coding; if not, 1.2 was used, and the process was stopped.
2. Software citation 2.1 Cited 2.2 Non-cited	This item indicates whether the author(s) of an article used and cited software. If software was used and cited, then it was encoded as 2.1; otherwise, it was encoded as 2.2.
3. Software section 3.1 Y 3.2 N	This item refers to whether the article included a section/subsection titled "Software" or "Software citation" or had a software-related section.
4. Software metadata collection 4.1 Collection of software metadata by the author 4.2 Using public software metadata set	This item refers to the source of the software metadata. If the author(s) of the article created and used software citations, it was encoded as 4.1; if the author(s) obtained the software from a publicly accessible source, it was encoded as 4.2.

5. Software reference types 5.1 Citation 5.2 DOI 5.3 URL 5.4 In-text name 5.5 No name	This item refers to whether the software used can be easily traced. If the article included software citations, it was encoded as 5.1; if a DOI was provided, 5.2 was used; and 5.3 was used if the URL was included. If a specific name was used for tracking, it was encoded as 5.4; if there was no name, it was encoded as 5.5. **Note:** These codes are not mutually exclusive; an article may provide both a DOI and software names.
6. Section referencing the software used in the study 6.1 References 6.2 Title 6.3 Abstract 6.4 Keywords 6.5 Acknowledgments 6.6 Methodology	This item identifies the section of the paper that mentions software. For example, if the abstract and methodology sections mentioned software, these mentions were encoded as 6.3 and 6.6, respectively.
7. Types of archive 7.1 Commerce 7.2 Institution 7.3 Government 7.4 Journal 7.5 Others *7.6 Personal statement	This item refers to the type of software archive mentioned.
8. Software accessibility 8.1 Inaccessible 8.2 Purchased access 8.3 Free access *8.4 Software available on request	This item refers to whether the public can access the software: 8.1 indicates that the software cannot be accessed, for example where an article mentioned that some software was used but did not provide any access points; 8.2 indicates that the software used in the article provides purchase access rights only; 8.3 means that the software is free to access; and 8.4 shows that the software provided in the article is available upon request.

Table 2. Coding scheme (Own editing)

The credibility of content analysis results depends on the availability of sufficient, appropriate, and highly saturated data. Therefore, data collection, analysis, and reporting of results go hand in hand. Improving the credibility of content analysis begins with comprehensive preparation before research and development of advanced skills in data collection, credibility discussion, and reporting of results (Pölkki et al. 2014). For example, when an article refers to a database, it is difficult to determine whether the article is merely mentioning the database or has collected data from it. Coding can avoid this problem by conveying the way in which the researchers reached their insights. For example, when the code generates the text (Figure 1), we can see whether the article used the software under study (code 1.1) or not (code 1.2).

4. Results

Based on the 271 articles we searched and the coding strategies listed above, we manually calculated the number of articles in each coding category through content analysis. Figure 1 shows that approximately 80% of the articles (216) used software in their research, and the remaining 20% (55) did not.

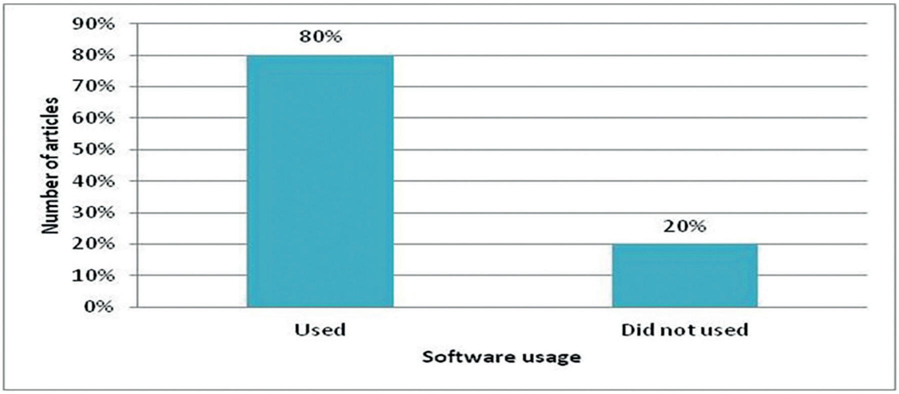

Figure 1. Use of software in surveyed papers (Own editing)

For the 216 articles that used software, only 15.28% of them cited the software used, thus recording the metadata about the software used via references; 4.17% of them recorded the URL of the website from which the software was obtained, 50.93% of them provided the DOI of the software used, and 29.63% of them mentioned the name of the relevant software within the study. We can see that there is no uniform standard for citing software, and citation formats vary. Some articles provided contact information about software developers or copyright owners, but the researchers did not adequately value and recognize the contribution of those software developers because they often neglected to cite the software they used in the article, which affects or even hinders the sustainable use and reuse of software.

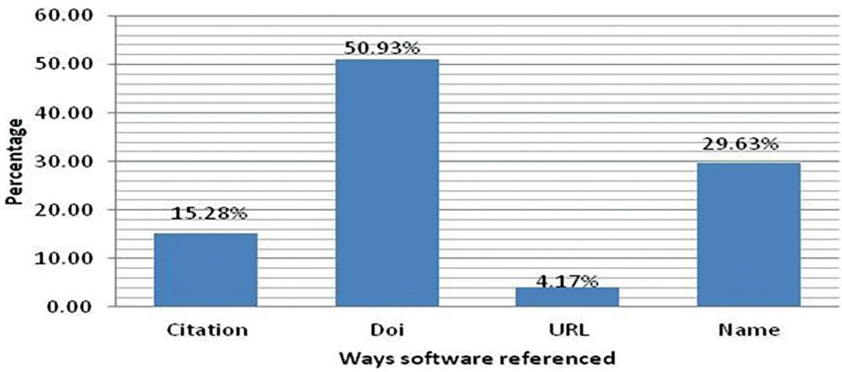

Figure 2. How software was referenced in the surveyed articles (Own editing)

Moreover, approximately 43% of the 216 articles included in their methodology section information about the software used. After that, 15% of articles mentioned software information in other sections; 17% mentioned it in the abstract; 15% mentioned it in the acknowledgments; 4% mentioned it in the keywords section; and 3% provided it in the title. The survey results are as shown in Figure 3.

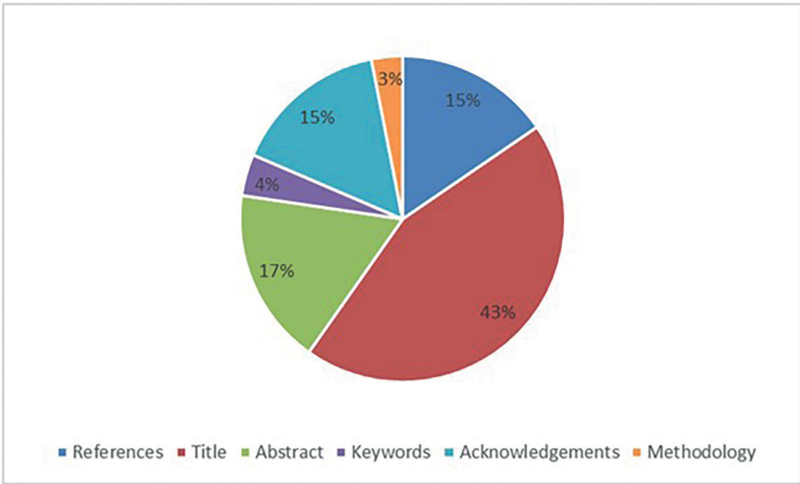

Figure 3. Article sections mentioning software information (Own editing)

According to the survey, different source types of research software exist (see Figure 4). Commercial software accounts for approximately 52% of the total. Software from government and other noncommercial institutions accounts for 4% and 9%, respectively. Approximately 20% of the software comes from specific journals that are for-profit or nonprofit. Moreover, personal website software metadata account for only 3%. The software source and archives are essential to software preservation; an exact copy and proper storage of the software can ensure its use or reuse in the future.

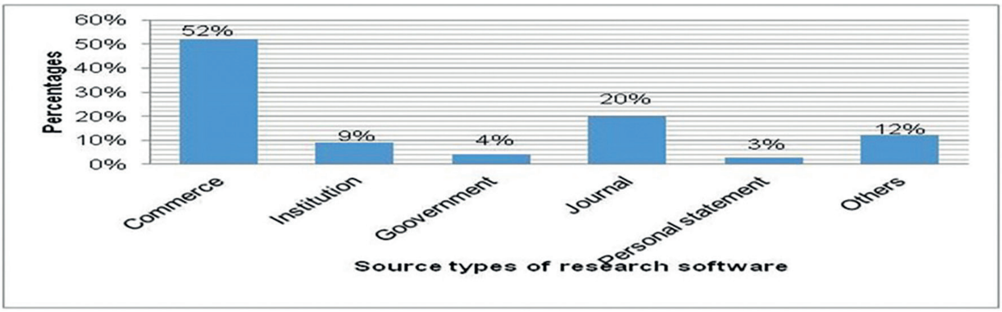

Figure 4. Source type of software used in research (Own editing)

Figure 5 shows the types of sources from which software was obtained in various disciplines. In earth science, agricultural science, computer science, and biology, 80%, 60%, 50%, and 35%, respectively, of the software used in articles can be accessed free. In medical and health sciences (71%) and social sciences (56%), it is difficult to access the software referenced in the articles. In management and education and educational research, most software is not freely accessed or used. Overall, open access to research software is minimal.

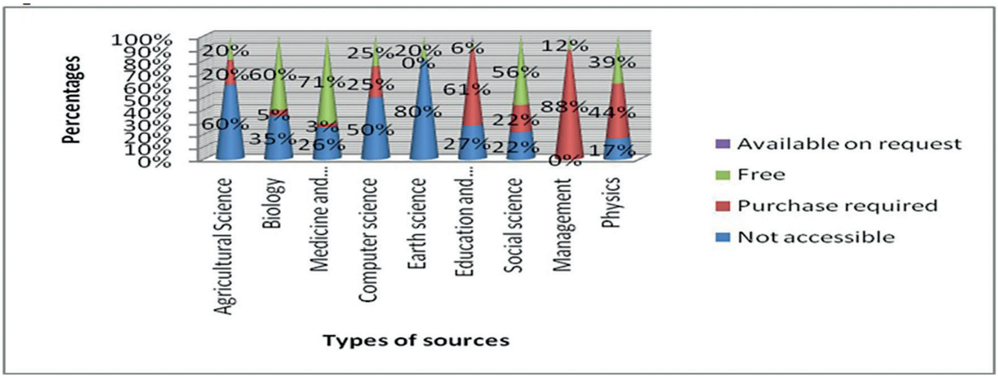

Figure 5. The accessibility types of software (Own editing)

Our findings indicate that 93% of software citations mention the name of the software; 66% provide an online publication source; 60% include the software accessed date; and 49%, 36%, 34%, 14%, and 9% indicate the place of publication, author, URL, publisher, and contributor of the software, respectively. The survey results (see Figure 6) show that in most of the articles the information about the software that was used is incomplete. Most of the software citations mentioned indicates only one or a few accessible types of software; this leads to challenges in tracking how software is used in research.

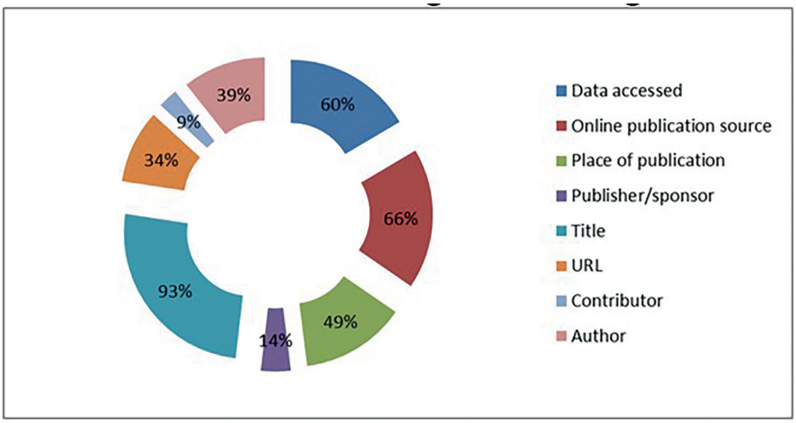

Figure 6. The information recorded in software citations (Own editing)

5. Discussion

According to the above analysis, there are some shortcomings in software citation practices. In this section, we will discuss our findings and the strategies that may enhance the sustainability of software citation practices to improve the quality of scientific research.

The evidence obtained through this paper clearly shows that software citation practices are diverse, and there are substantial problems in implementing these practices. From Figures 3 and 6, we can see that the ways of mentioning and citing software vary among the surveyed papers. From the perspective of scientific communication, these different standards of referring to software are certainly better than no reference, but they often fail to completely convey much of the relevant information, which reduces the use of specific software and negatively impacts software citation sustainability. Figure 3 shows that 43% of the surveyed articles provided software information in the methodology section, i.e., researchers used the software in the research process, but most of the articles lacked clarity on the creators of such software. Figure 6 shows that the name of the software appeared in 93% of the articles, which indicates that there is little information on the software used and that the information provided could provide little help in the development of software for users who are interested.

Though software is often cited in articles, the content and the placement of citations need to be revised and standardized. Moreover, unlike articles, software often changes over time, and the ability to find the specific version used in a given study is critical. Software citation practices present many common challenges, as data citation sustainability requires both practice and design innovations. The use and the reuse of software are essential for contemporary scientific research, and software citation practices need to be fully and consistently involved in the validation, replication, and construction of studies. Therefore, obstacles to sustainable software citation include the lack of citation standards, the difficulty of version control, etc. (Howison and Bullard 2016).

Software citation standards could be improved via cultural and technical solutions. An example of cultural change is the promotion of free open licensing of scientific software, which includes the improvement of code-sharing infrastructure and system design in measuring software contributions. One technical solution is to evaluate current practices and compare them with previously defined citation standards. With the technical challenges and relative novelty of these practices in scholarly communication, it is appropriate to make standards and guidelines for accessible software. Publishing software based on open-source policies, including links to its source sites, is a reliable way to build valuable resources for noncommercial researchers around the world (Huanget al. 2017). The accessibility of packages/codes is an essential element to facilitate their independent reproduction, verification, and usability. Hence, the reward system is almost entirely based on research publications, not the software that generates inspiring ideas during the research process, and the degree of recognition given to software is not proportional to the importance of the person who introduces or develops the software (Goble 2014).

Public and private organizations should perform software citation training to cultivate awareness in researchers and users. Software skills inculcated at the start of a research career would usually be used continuously throughout the whole research career. Additionally, strict peer review of software citations is very important for their sustainability. Addressing the sustainability of software used in the sciences includes many themes, from developing clear citation practices and building a community of reviewers to making existing credit and citation ecosystems pertaining to software more available.

6. Conclusions

We intended to find solutions to developing sustainable software citation practices to improve the quality of scientific research. Our findings show that the use of software does not match the citations or other references contained in the articles surveyed. Software citations are different from other elements of research output in scientific articles. They serve as an artifact, a tool, an agreement, and occasionally a publication, and they are the focus of ongoing activities.

Moreover, we confirmed our hypotheses about citation practices by focusing on their various outcomes. After coding and analyzing the results, we found that some articles entirely failed to mention software or had no section relating to the software used. Additionally, we discovered that most of the software information that was included was placed in the methodology section (43% compared with other sections of the articles), but most articles failed to mention the creators of such software. Additionally, we investigated the software information in the articles, identified a shortage of software information, and found to be inadequate the information on how the software was used, reused, modified, and reproduced in such a way as to enhance the development of academic research methods. The results indicated that most of the software archives are commercial; the very small amount of open-source software has limited the development and continuity of software citations in scientific research.

Furthermore, we found that there is no clear standard practice in software citation; we advocate that auditors/reviewers should take action to evaluate the correctness and practicality of citations. The repeatability of the results presented in the submission needs to be particularly emphasized. In short, software is both an artifact and a practice; it is a tool for both developing and simplifying research, which makes software practices quite complicated. Nonetheless, how software is implemented also provides an opportunity to address the issues raised in this article, which will greatly increase the efficiency of academic exchanges and improve practices in scientific publications. Our findings show improvements in software citation standards, rules and guidelines for software accessibility, knowledge about software citations, incentives in software development, training strategies, and the functions of reviewers as the means to influence and develop sustainable software citation practices to help compensate for deficiencies within scientific research. To address the sustainability of software citations, various objectives from the perspective of development and the community of research stakeholders are essential to improving the availability of software credit and citation ecosystems.

In a scientific study, it is necessary to foster collaboration among innovators, influencers, and users in the context of uniform software reference standards, which will enhance the understanding, recognition, and continuity of software citation practices. Different social entities such as governments, public organizations, private institutions, and other stakeholders need to work together to implement software citation guidelines.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Abbott, Alison, David Cyranoski, Nicola Jones, Brendan Maher, Quirin Schiermeier, and Richard Van Noorden. "Do metrics matter? Many researchers believe that quantitative metrics determine who gets hired and who gets promoted at their institutions. With an exclusive poll and interviews, Nature probes to what extent metrics are really used that way."*Nature*465, no. 7300 (2010): 860–863.

American Astronomical Society."Policy Statement on Software."(January 1, 2016). https://journals.aas.org/policy-statement-on-software

Bangerth, Wolfgang, Denis Davydov, Timo Heister, Luca Heltai, Guido Kanschat, Martin Kronbichler, Matthias Maier, Bruno Turcksin, and David Wells. "The deal. II library, version 8.4." *Journal of Numerical Mathematics*24, no. 3 (2016): 135–141. https://www.degruyter.com/document/doi/10.1515/jnma-2016-1045/html

Borgman, Christine L., Jillian C. Wallis, and Matthew S. Mayernik. "Who's got the data? Interdependencies in science and technology collaborations."*Computer Supported Cooperative Work (CSCW)*21, no. 6 (2012): 485–523. https://link.springer.com/article/10.1007/s10606-012-9169-z

Gambardella, Alfonso, and Bronwyn H. Hall. "Proprietary versus public domain licensing of software and research products."*Research Policy*35, no. 6 (2006): 875–892. https://www.sciencedirect.com/science/article/abs/pii/S0048733306000643

Goble, Carole. "Better software, better research."*IEEE Internet Computing*18, no. 5 (2014): 4–8.

Heinle, Albert, Wolfram Koepf, and Wolfram Sperber. "Some steps to improve software information." *ACM Communications in Computer Algebra* 51, no. 1 (2017): 1–11. https://dl.acm.org/doi/abs/10.1145/3096730.3096731

Howison, James, and James D. Herbsleb. "Incentives and integration in scientific software production."In*Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, 459–470.New York, United States: Association for Computing Machinery, 2013. https://dl.acm.org/doi/abs/10.1145/2441776.2441828

Huang, Xunhe, Jieru Xie, Newton O. Otecko, and Minsheng Peng. "Accessibility and update status of published software: Benefits and missed opportunities."*Frontiers in Research Metrics and Analytics*2 (2017):1.
https://www.frontiersin.org/articles/10.3389/frma.2017.00001/full

Hwang, Lorraine, Allison Fish, Laura Soito, MacKenzie Smith, and Louise H. Kellogg. "Software and the scientist: Coding and citation practices in geodynamics."*Earth and Space Science* 4, no. 11 (2017): 670–680.
https://agupubs.onlinelibrary.wiley.com/doi/full/10.1002/2016EA000225

Ince, Darrel C., Leslie Hatton, and John Graham-Cumming. "The case for open computer programs."*Nature*482, no. 7386 (2012): 485–488.
https://www.nature.com/articles/nature10836?em_x=22

Katz, Daniel S., and Neil P. Chue Hong. "Software citation in theory and practice."In *International Congress on Mathematical Software*, 289–296. Cham,Switzerland:Springer,2018.
https://link.springer.com/chapter/10.1007/978-3-319-96418-8_34

Knepley, Matthew G., Jed Brown, Lois Curfman McInnes, and Barry Smith. "Accurately citing software and algorithms used in publications."In *Workshop towards Sustainable Software for Science: Practice and Experiences*(WSSSPE). Denver, 2013.
http://dx.doi.org/10.6084/m9.figshare.785731

Krippendorff, Klaus. *Content Analysis: An Introduction to Its Methodology* (2nd ed.). Thousand Oaks, CA: Sage Publications,2004.
https://journals.sagepub.com/doi/abs/10.1177/1094428108324513?journalCode=orma

Mayernik, Matthew S., David L. Hart, Keith E. Maull, and Nicholas M. Weber. "Assessing and tracing the outcomes and impact of research infrastructures."*Journal of the Association for Information Science and Technology* 68, no. 6 (2017): 1341–1359.
https://asistdl.onlinelibrary.wiley.com/doi/full/10.1002/asi.23721

Mooney, Hailey, and Mark P. Newton. "The anatomy of a data citation: Discovery, reuse, and credit."Journal of Librarianship and Scholarly Communication1, no.1 (2012):eP1035.
https://doi.org/10.7710/2162-3309.1035

Nightingale, Julie M., and Gill Marshall. "Citation analysis as a measure of article quality, journal influence and individual researcher performance."*Radiography*18, no. 2 (2012): 60–67.
https://www.sciencedirect.com/science/article/abs/pii/S1078817411001374

O'Connor, Cliodhna. "'Appeals to nature'in marriage equality debates: A content analysis of newspaper and social media discourse." British Journal of Social Psychology 56, no. 3 (2017): 493-514.

Park, Hyoungjoo, and Dietmar Wolfram. "Research software citation in the Data Citation Index: Current practices and implications for research software sharing and reuse."*Journal of Informetrics*1, no. 32 (2019): 574–582.
https://www.sciencedirect.com/science/article/abs/pii/S1751157718302372

Pölkki, Tarja, Outi Kanste, Maria Kääriäinen, Satu Elo, and Helvi Kyngäs. "The methodological quality of systematic reviews published in high-impact nursing journals: A review of the literature." *Journal of Clinical Nursing* 23, no.3–4 (2014): 315–332.
https://onlinelibrary.wiley.com/doi/abs/10.1111/jocn.12132

Sahel, José-Alain. "Quality versus quantity: Assessing individual research performance."*Science Translational Medicine* 3, no. 84 (2011): 1–4.
https://www.science.org/doi/abs/10.1126/scitranslmed.3002249

Saldaña, Johnny. "First cycle coding methods." The coding manual for qualitative researchers (2009): 45-145.

Semetko, Holli A., and Patti M. Valkenburg. "Framing European politics: A content analysis of press and television news."*Journal of Communication* 50, no.2 (2000): 93–109. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1460-2466.2000.tb02843.x

Silvello, Gianmaria. "Theory and practice of data citation."*Journal of the Association for Information Science and Technology*69, no.1 (2018): 6–20. https://asistdl.onlinelibrary.wiley.com/doi/10.1002/asi.23917

Sugimoto, Cassidy R., Sam Work, Vincent Larivière, and Stefanie Haustein. "Scholarly use of social media and altmetrics: A review of the literature." *Journal of the Association for Information Science and Technology* 68, no. 9 (2017):2037–2062. https://asistdl.onlinelibrary.wiley.com/doi/10.1002/asi.23833