arXiv:2309.14571v2 [hep-ex] 4 Jan 2024

Software Citation in HEP: Current State and Recommendations for the Future

Matthew Feickert [1],*, *Daniel* S. Katz [2], *Mark* S. Neubauer [2],
Elizabeth Sexton-Kennedy [3], and *Graeme* A. Stewart [4]

[1]University of Wisconsin-Madison, Madison, Wisconsin, USA
[2]University of Illinois Urbana-Champaign, Illinois, USA
[3]Fermi National Accelerator Laboratory, Illinois, USA
[4]CERN, Switzerland

Abstract. In November 2022, the HEP Software Foundation and the Institute for Research and Innovation for Software in High-Energy Physics organized a workshop on the topic of Software Citation and Recognition in HEP. The goal of the workshop was to bring together different types of stakeholders whose roles relate to software citation, and the associated credit it provides, in order to engage the community in a discussion on: the ways HEP experiments handle citation of software, recognition for software efforts that enable physics results disseminated to the public, and how the scholarly publishing ecosystem supports these activities. Reports were given from the publication board leadership of the ATLAS, CMS, and LHCb experiments and HEP open source software community organizations (ROOT, Scikit-HEP, MCnet), and perspectives were given from publishers (Elsevier, JOSS) and related tool providers (INSPIRE, Zenodo). This paper summarizes key findings and recommendations from the workshop as presented at the 26th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2023).

1 Introduction

Software is a research product — an asset created as a byproduct of scientific research — that is ubiquitously used in and necessary to physics research, though it is not always given the same levels of importance and scholarly weight as other research products like publications and data products [1]. In November 2022, the HEP Software Foundation (HSF) and the Institute for Research and Innovation for Software in High-Energy Physics (IRIS-HEP) [2, 3] organized a topical workshop on software citation and recognition in the field of high energy physics (HEP) [4, 5]. The goal of the workshop was to provide a community discussion around ways in which HEP experiments handle citation of software and recognition for software efforts that enable physics results disseminated to the public. The workshop participants and primary presentations were from the LHC experiments that are primary stakeholders in IRIS-HEP operations: ATLAS, CMS, and LHCb; the particle physics open source software development communities: ROOT Team, Scikit-HEP [6], MCnet, and IRIS-HEP; as well as

---

*Corresponding author e-mail: matthew.feickert@cern.ch

the scientific publishing community and ecosystem most involved with HEP: Elsevier, the Journal of Open Source Software (JOSS) [7], and INSPIRE [8].

The principles of software citation that the HEP community is interested in engaging with are those established by the FORCE11 Software Citation working group [9]. These principles are defined as:

1. **Importance**: Software should be considered a legitimate and citable product of research. Software citations should be accorded the same importance in the scholarly record as citations of other research products, such as publications and data; they should be included in the metadata of the citing work, for example in the reference list of a journal article, and should not be omitted or separated. Software should be cited on the same basis as any other research product such as a paper or a book, that is, authors should cite the appropriate set of software products just as they cite the appropriate set of papers.

2. **Credit and Attribution**: Software citations should facilitate giving scholarly credit and normative, legal attribution to all contributors to the software, recognizing that a single style or mechanism of attribution may not be applicable to all software.

3. **Unique Identification**: A software citation should include a method for identification that is machine actionable, globally unique, interoperable, and recognized by at least a community of the corresponding domain experts, and preferably by general public researchers.

4. **Persistence**: Unique identifiers and metadata describing the software and its disposition should persist — even beyond the lifespan of the software they describe.

5. **Accessibility**: Software citations should facilitate access to the software itself and to its associated metadata, documentation, data, and other materials necessary for both humans and machines to make informed use of the referenced software.

6. **Specificity**: Software citations should facilitate identification of, and access to, the specific version of software that was used. Software identification should be as specific as necessary, such as using version numbers, revision numbers, or variants such as platforms.

Today the global research community now has these principles, citation policies from journal publishers, and modern open source tooling to facilitate the generation of software citations. There has also been growing movement among research software developers, research paper authors, and journal reviewers and editors [7] towards an increase in software citation. For the HEP community it is important to understand the current state (as of 2023) of software citation norms and culture in the field and how its importance can be conveyed and supported through community tooling, standards, and practices.

2 Current State of Software Citation in HEP

2.1 LHC Experiments

To understand the current state of software citation in the field reports from the ATLAS, CMS, and LHCb experiments were given that summarized the experiments' current standards and practices and future plans. ATLAS takes the approach of using a "catch-all" citation of all ATLAS software and firmware through the citation of an ATLAS public note that "briefly

describes the software and provides links to dynamic and persistent repositories wherein the code resides" [10]. This public note is then cited in many ATLAS papers. ATLAS additionally cites the paper for the ATLAS detector simulation software [11] as well as GEANT4 [12], and the Monte Carlo simulation generators used [13–16]. In terms of statistical analysis ATLAS cites the methodology papers that describe the techniques used in analyses, but in general does not cite the actual software that implements the techniques, with the notable exception of machine learning libraries [17, 18]. Citation practices are not uniformly consistent in the experiment though, with some physics groups beginning to regularly cite statistical libraries that provide clear citation guidelines [19, 20] (Principles 1 and 2).

CMS similarly has an established culture of regularly and consistently citing Monte Carlo generators, GEANT4, and machine learning tools. However, they note there could be improvement in the citation of the software that CMS itself produces, both in experimental internal notes and documentation as well as scientific publications. CMS also expressed positive views towards starting practices of publishing papers — either as CMS Collaboration publications or as limited authorship papers from the CMS Software and Computing Group — on CMS software, bringing with it increased visibility of scientific software development, documentation standards, and references of software version information (Principles 1 and 2).

LHCb has taken a more proactive stance on software citation following recommendations presented at the CHEP 2018 Conference [21] by providing an internal LHCb software citation starting template for software commonly used in analysis. Analysis teams are then encouraged to revise the template with the citations of the software used in their analysis with the goal that all high-level software used is properly cited (Principles 1, 2, and 6). These practices are encouraged in the collaboration, but not explicitly required, and so analysis teams may require citation guidelines to be provided. LHCb also noted that the citation practices of the HEP community are largely due to cultural norms rather than technical challenges, and that while LHCb strives to be citing more software in the future having LHC community recommendations on software citation would be useful for motivating better practices.

2.2 Software Projects

Views from prominent open source software projects and software communities inside of HEP were also discussed, with a broad range of community cultural views and practices. The ROOT team noted they explicitly are not interested in ROOT's software citation, as the ROOT team does not view it as adding value to their work, that updating citation information would require additional effort, and in the team's view the current HEP culture of citation with journal publications for larger software projects is working well. The ROOT team was careful to note though that these views are specifically limited to software citation for ROOT [22] and should not be viewed as being universal. In contrast, the Scikit-HEP community project has prioritized adopting software citation recommendations and tooling from the broader scientific open source community (e.g. Zenodo [23], `CITATION.cff` files [24]) to provide credit to the developers producing community tools (Principle 2) as well as recognize project contributions of multiple types [25]. Scikit-HEP views software citation as important to their community and would welcome HEP community guidelines to guide users of the community tools to easily and correctly cite the software. The MCnet community noted that as a community of Monte Carlo generator software projects they have benefited from consistent citation by the LHC experiments. Several community factors lead to this culture, including the MCnet community becoming organized in the leadup to the start of the LHC and providing clear citation guidelines and often making programmatic citation information available from the software itself. MCnet raised the potential problems with the current citation model of citing

papers for large releases of the software as this does not equally value or reward the development and maintenance labor that occurs between the long intervals between publications. As a result, MCnet is interested in both technical solutions as well as community guidelines and policy regarding software citation.

2.3 Publishing Community

Following the state of software citation in the HEP community, views and recommendations from INSPIRE, Elsevier, and JOSS were shared given their different roles related to scientific publishing and citation. INSPIRE is an integral part of how HEP interacts with publications, related metadata, and acquires updated citation information as tracked submissions move from preprints to publication. Having these capabilities for the citation information for software in HEP would be a technical boon. While INSPIRE currently only handles software papers, there are plans to add support for data products and software in the future, initially by harvesting metadata from relevant trusted repositories (e.g. INSPIRE HEP Zenodo community, HEPData, CERN Open Data portal). This information would be gathered by software digital object identifier (DOI), and could be aggregated across multiple releases of the same software. It is therefore important that software projects that seek citations in the future provide DOIs now (Principles 3, 5, and 6). Elsevier noted that it is the responsibility of the scientific community to reach a consensus on how to cite software and to share these guidelines with publishers, which can then better instruct journal editors and referees what the expectations for citation are and how to support them. JOSS noted that in addition to incentivizing high quality research software with the journal guidelines and review standards, JOSS can also help bridge the cultural and technical gaps between traditional publication citation and the citation of software directly.

3 Recommendations

In addition to establishing guidelines for the HEP community, providing recommendations of software citation best practices and supported tooling aids in community adoption of new guidelines. A behavior step that can be implemented is for software projects to clearly document a recommended citation and have this information be easily findable anywhere the software source code or distributions are hosted or documented (e.g. version control repositories, public documentation websites, package indexes, archives). There has been historical precedent in HEP for tools to provide recommendations for how to cite the software being used by printing it as a runtime banner to standard output, as seen in Listing 1. This method was developed before citation conventions were established more broadly in the scientific computing community, and modern practices would generally avoid interrupting user logs with this information. It is instead preferable, in addition to having a clearly documented and advertised recommended citation, to provide citation APIs in the software — both at the language level and at the command line interface if the software supports one.

In addition to having clear citation recommendations, it is beneficial to adopt a standardized citation file format. A strong choice is the recent Citation File Format [24] which is serialized as YAML as a `CITATION.cff`, as seen in Listing 2. `CITATION.cff` files have the benefit of being both human- and machine-readable with a well defined, versioned schema. Through related tooling `CITATION.cff` can also be programmatically validated against schemas and converted to other citation formats (e.g., BibTeX, CodeMeta, EndNote, RIS, schema.org, Zenodo, APA). `CITATION.cff` also benefits through supported integration

```
 *-------------------------------------------------------------------------*
 |                                                                         |
 | *---------------------------------------------------------------------* |
 | |                                                                     | |
 | |                                                                     | |
 | |   PPP   Y   Y  TTTTT  H   H  III    A       Welcome to the Lund Monte Carlo!  | |
 | |   P  P   Y Y     T    H   H   I    A A      This is PYTHIA version 8.230      | |
 | |   PPP     Y      T    HHHHH   I   AAAAA     Last date of change:  6 Oct 2017  | |
 | |   P       Y      T    H   H   I   A   A                                       | |
 | |   P       Y      T    H   H  III  A   A     Now is 06 May 2023 at 01:12:28    | |
 | |                                                                     | |
 ...
 | |   The main program reference is 'An Introduction to PYTHIA 8.2',    | |
 | |   T. Sjostrand et al, Comput. Phys. Commun. 191 (2015) 159          | |
 | |   [arXiv:1410.3012 [hep-ph]]                                        | |
 | |                                                                     | |
 | |   The main physics reference is the 'PYTHIA 6.4 Physics and Manual',  | |
 | |   T. Sjostrand, S. Mrenna and P. Skands, JHEP05 (2006) 026 [hep-ph/0603175]  | |
 | |                                                                     | |
 | |   An archive of program versions and documentation is found on the web:  | |
 | |   http://www.thep.lu.se/Pythia                                      | |
 | |                                                                     | |
 | |   This program is released under the GNU General Public Licence version 2.  | |
 | |   Please respect the MCnet Guidelines for Event Generator Authors and Users.  | |
 | |                                                                     | |
 ...
```

Listing 1: Sections of an example runtime banner printed to standard output from PYTHIA 8.2 [14] with citation guidelines — a historical community norm.

```
cff-version: 1.2.0
message: "If you use this software, please cite it as below."
authors:
  - family-names: Druskat
    given-names: Stephan
    orcid: https://orcid.org/0000-0003-4925-7248
title: "My Research Software"
version: 2.0.4
doi: 10.5281/zenodo.1234
date-released: 2021-08-11
```

Listing 2: Example of a minimal CITATON.cff file using v1.2.0 of the CFF schema [24].

with GitHub[1], Zenodo, and Zotero, allowing for the citation information to be reliably exported to multiple services from a single file (Principle 2). The integration with Zenodo is significant, as the HEP community is already frequent users of Zenodo for long term archival of source code (Principle 4) and DOI generation for the source code of software releases (Principle 3). Software projects that adopt the use of CITATION.cff and archive the source code with Zenodo have a clearly defined toolchain provenance for citation information dissemination (Principle 5). Given this, it is recommended that there is a single source of truth for citation information, such as a CITATION.cff file, that is under version control with the software source code and is used to generate all other metadata or forms of citation information by other services.

4 Conclusions

Revisiting the software citation principles in the view of current approaches and technologies in HEP provides a structure for starting community guidelines:

1. **Importance**: As a field HEP understands software is important, but improvements could be made on views towards software as a research product.

2. **Credit and Attribution**: The giving of credit is improving in HEP, but the community can leverage software friendly journals (e.g., JOSS) to help accelerate this.

---

[1]Providing a "Cite this repository" button on a repository with a CITATION.cff file.

3. **Unique Identification**: Use of Zenodo archives already exists in HEP, which provides well integrated tooling for DOI generation. The use of `CITATION.cff` files in software repositories can help as well.

4. **Persistence**: Zenodo provides long term archival of source code and project metadata.

5. **Accessibility**: HEP is becoming more FAIR [26, 27] focused, bringing with it an increased focus on accessibility. As `CITATION.cff` provides a common framework for metadata, adopting it as a community standard for software citation information allows for greater accommodation and discovery by citation discovery tools.

6. **Specificity**: Version numbers of software should be included in `CITATION.cff` files and the version used for analysis should be reported in publications.

It is seen there are both social and technical tooling challenges to be addressed to reach HEP community guidelines and recommendations for software citation. While there exist multiple practices towards software citation in the HEP community today, this should not be viewed as a large challenge towards global community standards adoption as variations in homogeneity of practice are common even in journal publication. The community wide agreement that software citation is important, should be practiced more often, and provides both social and technical benefits gives sufficient motivation to develop HEP community wide recommendations in the near future.

Acknowledgments

References

[1] K. Cranmer et al., SciPost Phys. **12**, 037 (2022), `2109.04981`

[2] P. Elmer, M. Neubauer, M.D. Sokoloff (2017), `1712.06592`

[3] *IRIS-HEP website*, http://iris-hep.org

[4] M. Feickert, D.S. Katz, M.S. Neubauer, E. Sexton-Kennedy, G.A. Stewart, *Software Citation and Recognition Workshop Report* (2023), forthcoming

[5] *Software Citation and Recognition in HEP*, https://indico.cern.ch/event/1211229/ (2022)

[6] E. Rodrigues et al., EPJ Web Conf. **245**, 06028 (2020), `2007.03577`

[7] A.M. Smith, K.E. Niemeyer, D.S. Katz, L.A. Barba, G. Githinji, M. Gymrek, K.D. Huff, C.R. Madan, A.C. Mayes, K.M. Moerman et al., PeerJ Computer Science **4**, e147 (2018)

[8] *Inspire*, `http://inspirehep.net`

[9] A.M. Smith, D.S. Katz, K.E. Niemeyer, PeerJ Computer Science **2**, e86 (2016)

[10] ATLAS Collaboration, *The ATLAS Collaboration Software and Firmware*, ATL-SOFT-PUB-2021-001 (2021), `https://cds.cern.ch/record/2767187`

[11] ATLAS Collaboration, Eur. Phys. J. C **70**, 823 (2010), `1005.4568`

[12] S. Agostinelli et al. (GEANT4), Nucl. Instrum. Meth. A **506**, 250 (2003)

[13] T. Sjöstrand, S. Mrenna, P.Z. Skands, Comput. Phys. Commun. **178**, 852 (2008), `0710.3820`

[14] T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C.O. Rasmussen, P.Z. Skands, Comput. Phys. Commun. **191**, 159 (2015), `1410.3012`

[15] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.S. Shao, T. Stelzer, P. Torrielli, M. Zaro, JHEP **07**, 079 (2014), `1405.0301`

[16] E. Bothmann et al. (Sherpa), SciPost Phys. **7**, 034 (2019), `1905.09127`

[17] F. Chollet et al., *Keras*, https://keras.io (2015)

[18] M. Abadi et al., *TensorFlow: Large-scale machine learning on heterogeneous systems* (2015), software available from tensorflow.org, https://www.tensorflow.org/

[19] L. Heinrich, M. Feickert, G. Stark, *pyhf: v0.7.4*, https://github.com/scikit-hep/pyhf/releases/tag/v0.7.4, https://doi.org/10.5281/zenodo.11697

[20] L. Heinrich, M. Feickert, G. Stark, K. Cranmer, Journal of Open Source Software **6**, 2823 (2021)

[21] D.S. Katz, *Software Citation at CHEP* (2018), CHEP 2018 Conference, https://indico.cern.ch/event/587955/contributions/3012261/

[22] R. Brun, F. Rademakers, Nucl. Instrum. Meth. A **389**, 81 (1997)

[23] *Zenodo*, https://zenodo.org/

[24] S. Druskat, J.H. Spaaks, N. Chue Hong, R. Haines, J. Baker, S. Bliven, E. Willighagen, D. Pérez-Suárez, O. Konovalov, *Citation File Format* (2021), https://doi.org/10.5281/zenodo.5171937

[25] *all-contributors*, https://github.com/all-contributors/all-contributors

[26] M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.W. Boiten, L.B. da Silva Santos, P.E. Bourne et al., Scientific Data **3**, 160018 (2016)

[27] N.P. Chue Hong, D.S. Katz, M. Barker, A.L. Lamprecht, C. Martinez, F.E. Psomopoulos, J. Harrow, L.J. Castro, M. Gruenpeter, P.A. Martinez et al., *FAIR Principles for Research Software version 1.0 (FAIR4RS Principles v1.0)* (2022), https://doi.org/10.15497/RDA00068