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

Matthew Feickert
(University of Wisconsin-Madison)
matthew.feickert@cern.ch
Daniel Katz, Mark Neubauer, Elizabeth Sexton-Kennedy, Graeme Stewart

International Conference on Computing in High Energy and Nuclear Physics (CHEP) 2023

May 8th, 2023

American Family Insurance
Data Science Institute
UNIVERSITY OF WISCONSIN–MADISON

iris hep

Co-conveners

Daniel Katz

University of Illinois at Urbana-Champaign/ NCSA

Mark Neubauer

University of Illinois at Urbana-Champaign

Elizabeth Sexton-Kennedy

FNAL

Graeme Stewart

CERN

Software Citation and Recognition Workshop

- 2022 HSF/IRIS-HEP Blueprint Process Workshop

  *This meeting aims 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.*

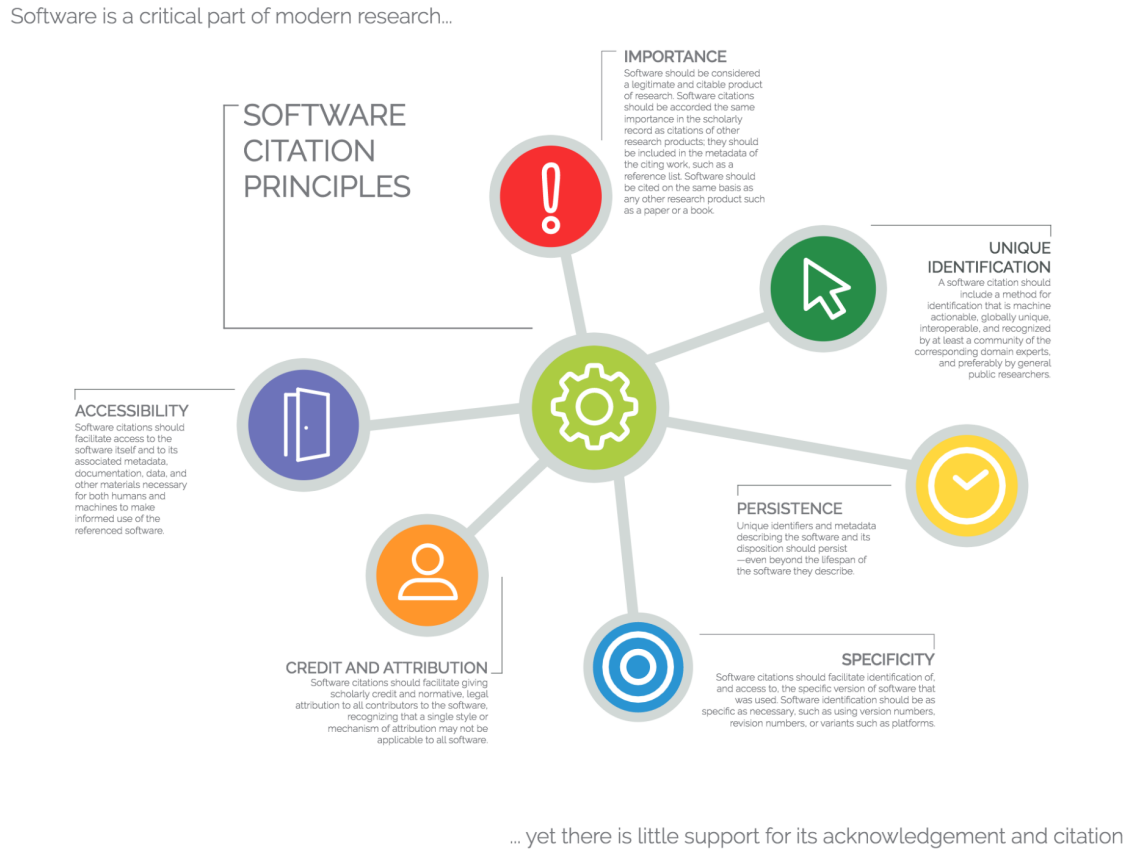
- Had representation from:
  - **Experiments**: ATLAS, CMS, LHCb
  - **Software project communities**: ROOT Team, Scikit-HEP, MCnet, IRIS-HEP
  - **Publishers**: INSPIRE, Elsevier, Journal of Open Source Software (JOSS)

This work was supported by the National Science Foundation under Cooperative Agreement OAC-1836650. NSF 1836650

Principles of Software Citation

As established by FORCE11 Software Citation working group (2016, DOI 10.7717/peerj-cs.86)

1. Importance
2. Credit and Attribution
3. Unique Identification
4. Persistence
5. Accessibility
6. Specificity



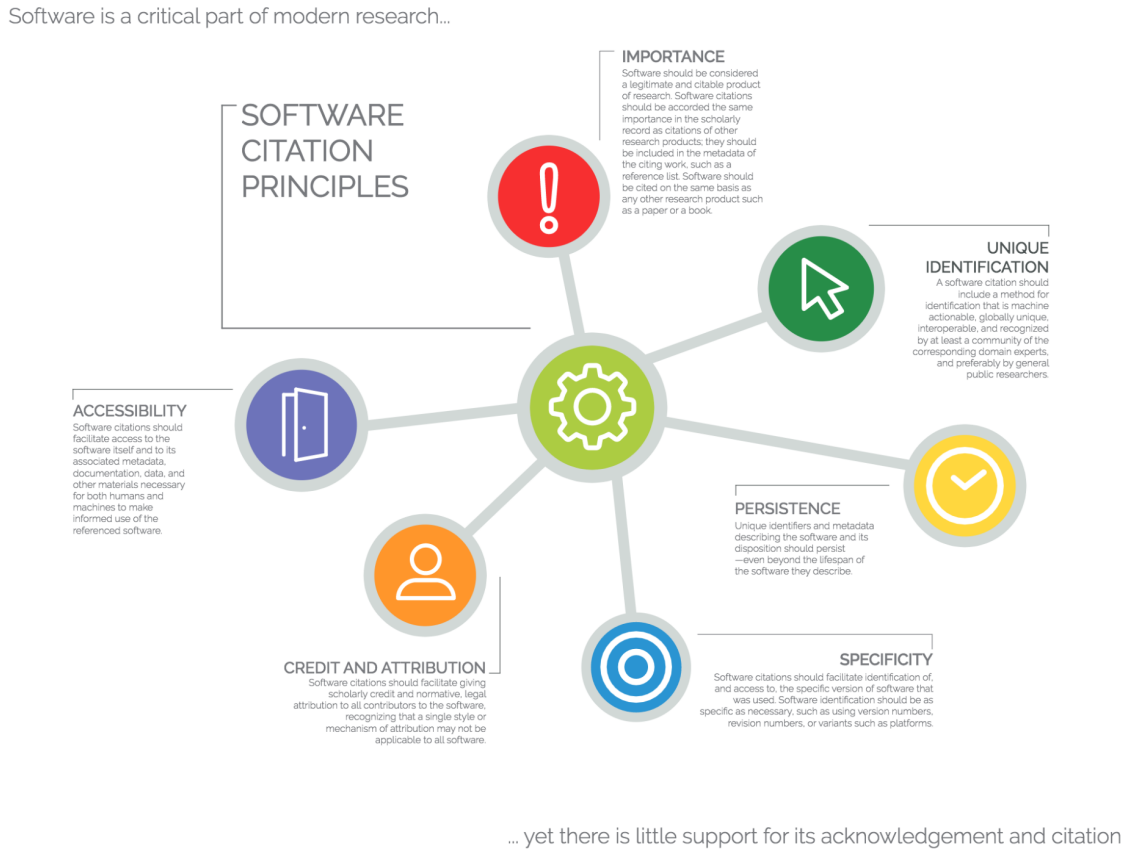
Software Citation Principles

image credit: Data Cite

Current State of Software Citation

Currently have
(in 2023)

- Software citation principles
- Policies from publishers
- Modern open source tooling
- Beginning of movement among developers, paper authors, journal reviewers and editors



Software Citation Principles

image credit: Data Cite

Current State of Software Citation in HEP: ATLAS & CMS

ATLAS

- Use a "catch-all" citation for ATLAS software tools
- For statistical analysis and ML generally cite the papers for the methods, but not the tools and software
  - Have seen some changes when the tools explicitly ask to be cited

CMS

- Endorses large CMS software projects having peer reviewed papers that would be cited in physics papers
- Have expressed positive views on additional papers being written and published

Current State of Software Citation in HEP: LHCb

- Following recommendations of Daniel Katz's CHEP 2018 presentation
- Most papers aim to cite all high-level software used in the analysis
  - Cite the software, and if there's a paper for the software cite that too
- Analysts are still adopting this habit and need reminding
  - Though open to idea: "We should cite software more"

Description	Ref. `cite code`
PYTHIA	[Sjostrand:2007gs,*Sjostrand:2006za]
EVTGEN	[Lange:2001uf]
PHOTOS	[davidson2015photos]
GEANT4	[Allison:2006ve, *Agostinelli:2002hh]
LHCb simulation	[LHCb-PROC-2011-006]
RapidSim	[Cowan:2016tnm]
DIRAC	[Tsaregorodtsev:2010zz,*BelleDIRAC]
sPlot	[Pivk:2004ty]
sFit	[Xie:2009rka]
BDT	[Breiman]
BDT training	[AdaBoost]
TMVA	[Hocker:2007ht,*TMVA4]
RooUnfold	[Adye:2011gm]
scikit-learn	[Scikit-learn-paper]
LAURA^{++}	[Back:2017zqt]
`hep_ml`	[Rogozhnikov:2016bdp]
`root_numpy`	[root-numpy]
`GammaCombo`	[GammaCombo]
TENSORFLOW	[tensorflow2015-whitepaper]
`ROOT`	[Brun, Rademaker]
`RooFit`	[Verkerke, Kirkby]
`scikit-hep`	[Scikit HEP]

LHCb citation template

Current State of Software Citation in HEP: Software Projects

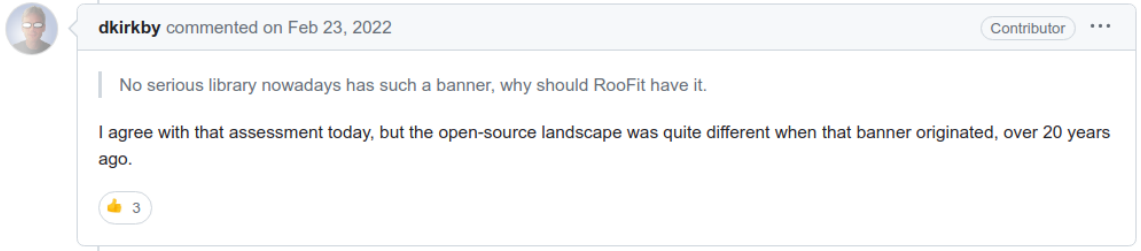
- Community views vary widely
  - **ROOT team**: Explicitly not interested in software citation *for ROOT* (view is too little impact)
    - "ROOT's opinion likely cannot be extrapolated"
  - **Scikit-HEP**: Adopting software citation recommendations from broader open source world community norms — interested in more citations
  - **MCnet**: Find Monte Carlo generators are broadly well-cited (point to LHC experiments regular citations) — current system working well
- Agreement on importance of technical solutions
  - Programmatic discovery of citations important

Recommendations from Journals and Publishers

- **INSPIRE**
  - Currently handles software **papers**, but has plans to add support for **Data and Software**
  - Citations would be tracked and counted **by DOI**
- **Elsevier**
  - **Community needs to reach consensus** on how to cite software, and share outcome with publishers (won't take lead)
  - Publishers can **better instruct editors and referees** what publishers expect from them
- **Journal of Open Source Software**
  - In addition to incentivizing high quality software, JOSS can help **bridge the gap**
  - "recognize that for most researchers, papers and not software are the currency of academic research"

Recommendations: Historical retrospective

- Software in the field might advertise citation/copyright information with runtime banners
- Conventions were not firmly established in the broader scientific computing community



- Interrupting user logs is now **avoided** given modern tooling. **Consider using APIs**.

```
# CLI API                  # Python API
$ mytool --citation        import mytool
$ mytool --cite            mytool.utils.citation()
```

```
#------------------------------------------------------------------------
#                         FastJet release 3.4.0
#                 M. Cacciari, G.P. Salam and G. Soyez
#     A software package for jet finding and analysis at colliders
#                           http://fastjet.fr
#
# Please cite EPJC72(2012)1896 [arXiv:1111.6097] if you use this package
# for scientific work and optionally PLB641(2006)57 [hep-ph/0512210].
#
# FastJet is provided without warranty under the GNU GPL v2 or higher.
# It uses T. Chan's closest pair algorithm, S. Fortune's Voronoi code,
# CGAL and 3rd party plugin jet algorithms. See COPYING file for details.
#------------------------------------------------------------------------
```

```
RooFit v3.60 -- Developed by Wouter Verkerke and David Kirkby
                Copyright (C) 2000-2013 NIKHEF, University of California & Stanford U
                All rights reserved, please read http://roofit.sourceforge.net/licens
```

```
 *-------------------------------------------------------------------------------*
 |                                                                               |
 |  *-------------------------------------------------------------------------*  |
 |  |                                                                         |  |
 |  |                                                                         |  |
 |  |   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. |
 |  |                                                                         |  |
...
```

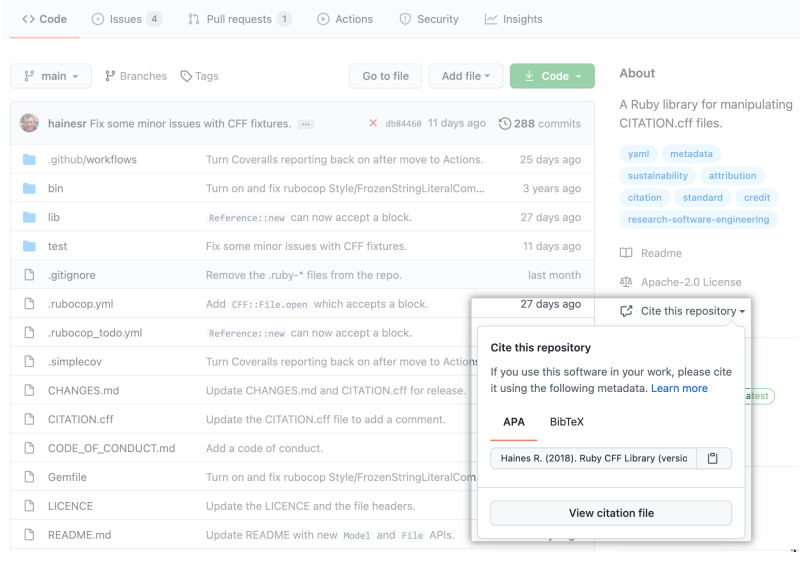
Recommendations: CITATION.cff

- Adopt the Citation File Format as a common standard and add a `CITATION.cff` to project repository
  - Human- and machine-readable file format in YAML
  - Has well defined, versioned schema
  - Convertible to other citation formats (BibTeX, CodeMeta, EndNote, RIS, schema.org, Zenodo, APA)
- Supported by GitHub, Zenodo, and Zotero!
- Web tool initializer for easily creating first `CITATION.cff`
- Tooling for validation

```
$ python -m pip install cffconvert
$ cffconvert --validate
Citation metadata are valid according to schema version 1.2.0.
```

```
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
```

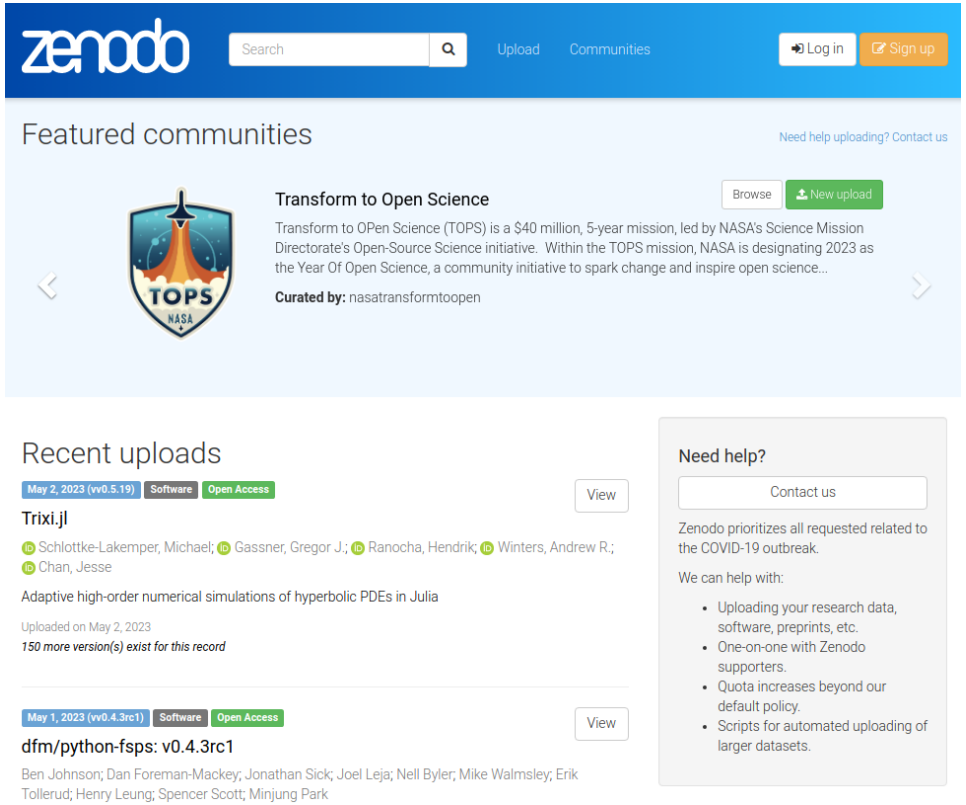
Example of minimal `CITATION.cff`



Recommendations: Zenodo

Versioned archive of **everything**: code, documents, data products, data sets

(See Lars Holm Nielsen's CHEP 2023 talk)



DOI for project and each version

Why use Zenodo?

- **Safe** — your research is stored safely for the future in CERN's Data Centre for as long as CERN exists.
- **Trusted** — built and operated by CERN and OpenAIRE to ensure that everyone can join in Open Science.
- **Citeable** — every upload is assigned a Digital Object Identifier (DOI), to make them citable and trackable.
- **No waiting time** — Uploads are made available online as soon as you hit publish, and your DOI is registered within seconds.
- **Open or closed** — Share e.g. anonymized clinical trial data with only medical professionals via our restricted access mode.
- **Versioning** — Easily update your dataset with our versioning feature.
- **GitHub integration** — Easily preserve your GitHub repository in Zenodo.
- **Usage statisics** — All uploads display standards compliant usage statistics

Recommendations: Make clear how to cite in docs

- The easiest, but least robust way: If you have a particular citation that you want people to use, put it **everywhere**
  - Version control repository README
  - Online software documentation (landing page, how to cite page)
  - Package distribution websites (e.g. PyPI)
- Have **single source of truth** for citations: version control repository that all other sources derive from.
- Make your **citation preferences clear** to the world and SEO. Do not rely on people emailing to ask (they shouldn't have to).

Use and Citations

Citation

The preferred BibTeX entry for citation of `pyhf` includes both the Zenodo archive and the JOSS paper:

```
@software{pyhf,
  author = {Lukas Heinrich and Matthew Feickert and Giordon Stark},
  title = "{pyhf: v0.7.1}",
  version = {0.7.1},
  doi = {10.5281/zenodo.1169739},
  url = {https://doi.org/10.5281/zenodo.1169739},
  note = {https://github.com/scikit-hep/pyhf/releases/tag/v0.7.1}
}

@article{pyhf_joss,
  doi = {10.21105/joss.02823},
  url = {https://doi.org/10.21105/joss.02823},
  year = {2021},
  publisher = {The Open Journal},
  volume = {6},
  number = {58},
  pages = {2823},
  author = {Lukas Heinrich and Matthew Feickert and Giordon Stark and Kyle Cranmer},
  title = {pyhf: pure-Python implementation of HistFactory statistical models},
  journal = {Journal of Open Source Software}
}
```

pyhf's "Use and Citations" page in documentation

Revisiting Software Citation Principles in HEP

- **Importance:** As a field HEP understands software is important, but improvements could be made on views towards research products
- **Credit and Attribution:** Improving in HEP, but can leverage software friendly journals (i.e., JOSS) to help this
- **Unique Identification:** Zenodo DOIs are common to HEP. CITATION.cff files can help as well.
- **Persistence:** Long term archival through Zenodo is common practice
- **Accessibility:** HEP is becoming more FAIR focused. CITATION.cff provides common framework for metadata. Further support of INSPIRE by field could provide strong database access.
- **Specificity:** Include version numbers in CITATION.cff

Summary

- Software citation is an **ongoing process** that is not straightforward (for any scientific field)
- Differing community **processes and standards** exist in HEP (we've never been homogeneous)
- Agreement that more citation is probably useful and **programatic discovery** of citations is important
- With modern tools and standards have the opportunity to **expand and standardize**
- Final summary paper from workshop forthcoming

Backup

CITATION.cff: Which DOIs?

Personal choice, but given specific release DOIs are generated *after* the release is made to have the versioned and distributed `CITATION.cff` be correct at release would recommend the Zenodo project level ("cite all versions") DOI.

See valid `CITATION.cff` in talk repository (`examples/CITATION.cff`):

```
cff-version: 1.2.0
message: "Please cite the following works when using this software."
type: software
authors:
- family-names: "Feickert"
  given-names: "Matthew"
  orcid: "https://orcid.org/0000-0003-4124-7862"
  affiliation: "University of Wisconsin-Madison"
title: "mylibrary: v1.2.3"
version: 1.2.3
# This is the _project_ DOI ("cite all versions" DOI on Zenodo page)
doi: 10.5281/zenodo.1234567
repository-code: "https://github.com/myorg/mylibrary/releases/tag/v1.2.3"
url: "https://mylibrary.readthedocs.io/en/v1.2.3/"
keywords:
  - example
  - software
```

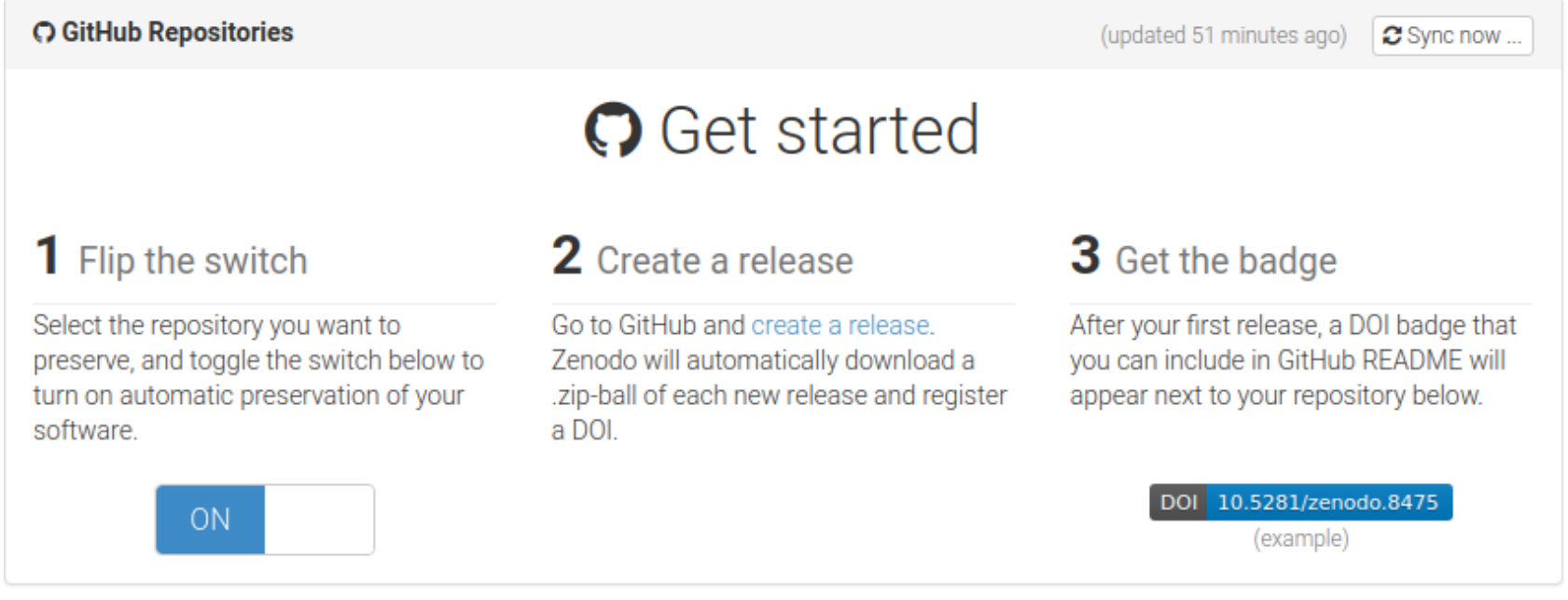
CITATION.cff: How to keep up to date?

- As plain text, very easy to update version information when cutting a release
- Can use tool control of version update to make it easier
  - Example: `tbump`
  - `$ tbump <version target>`
- Also possible to have automated version bump workflows using continuous integration
- (Jumping ahead a slide) What about the Zenodo DOI?
  - For simplicity, use the project level DOI and not the version level DOI

```
cff-version: 1.2.0
message: "Please cite the following works when using this software."
type: software
...
title: "mylibrary: v1.2.3"
version: 1.2.3
doi: 10.5281/zenodo.1123456
repository-code: "https://github.com/myorg/mylibrary/releases/tag/v1.2.3"
url: "https://mylibrary.readthedocs.io/en/v1.2.3/"
```

Zenodo: DOI minting made easy

- Everything on Zenodo has a DOI
  - Provides both a **project** DOI (resolves to latest) and **version specific** DOI
- Enable it to automatically preserve work from GitHub (can also directly upload, but lose out on automation)
  - Benefit from having a DOI for **every version** regardless of software paper landscape state
- Once you have a DOI, put it **everywhere** (again)
  - Recommend sharing the project DOI and letting users select a specific version if they want it

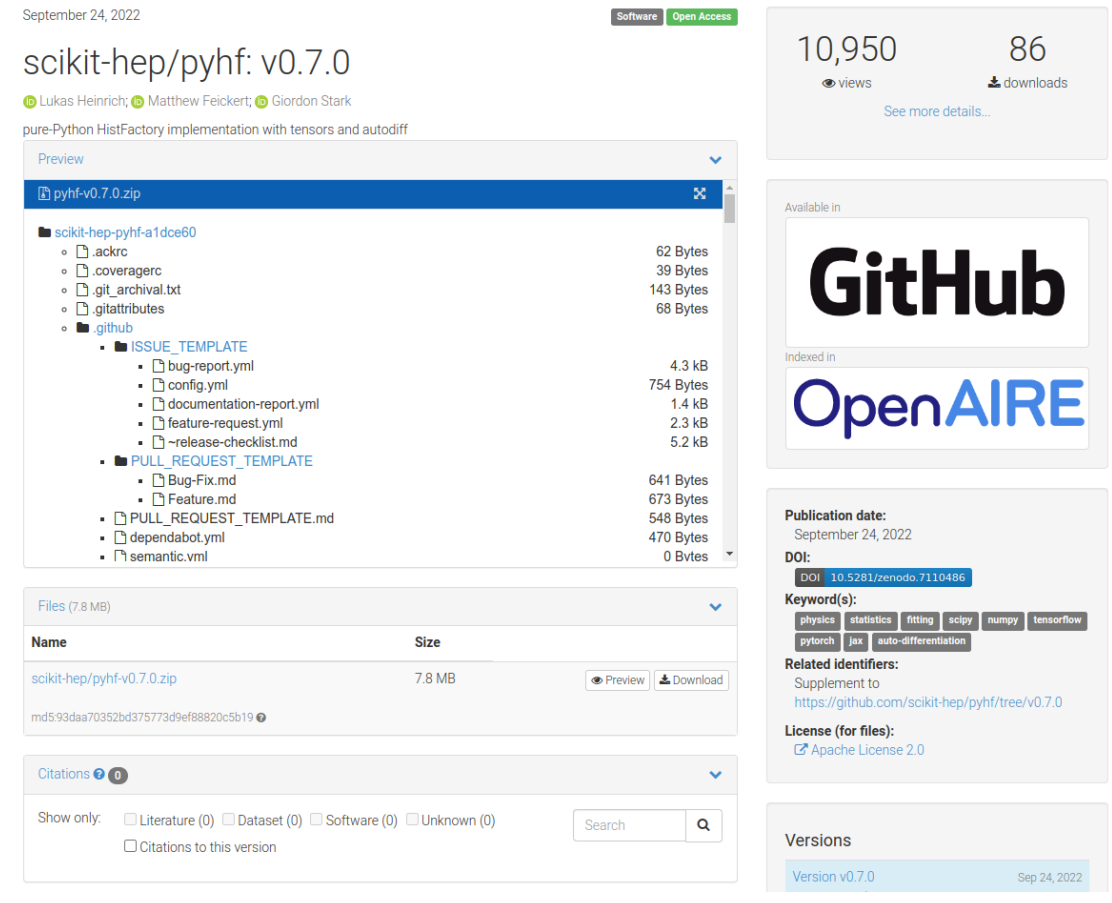


Zenodo + CITATION.cff

CITATION.cff used by Zenodo importer to fully define Zenodo archive metadata

71 lines (71 loc) · 2.47 KB

```yaml
cff-version: 1.2.0
message: "Please cite the following works when using this software."
type: software
authors:
- family-names: "Heinrich"
  given-names: "Lukas"
  orcid: "https://orcid.org/0000-0002-4048-7584"
  affiliation: "Technical University of Munich"
- family-names: "Feickert"
  given-names: "Matthew"
  orcid: "https://orcid.org/0000-0003-4124-7862"
  affiliation: "University of Wisconsin-Madison"
- family-names: "Stark"
  given-names: "Giordon"
  orcid: "https://orcid.org/0000-0001-6616-3433"
  affiliation: "SCIPP, University of California, Santa Cruz"
title: "pyhf: v0.7.0"
version: 0.7.0
doi: 10.5281/zenodo.1169739
repository-code: "https://github.com/scikit-hep/pyhf/releases/tag/v0.7.0"
url: "https://pyhf.readthedocs.io/en/v0.7.0/"
keywords:
  - python
  - physics
  - statistics
  - fitting
  - scipy
  - numpy
  - tensorflow
  - pytorch
  - jax
  - auto-differentiation
license: "Apache-2.0"
```



The end.