Research Software

Preservation & Attribution

Daina Bouquin

Center for Astrophysics | Harvard & Smithsonian

because you want credit for your work

because you care about science

I don't want you to have to think about citation.

I want you to have a scientific legacy.

Your work will be the foundation on which future generations must build an improved understanding of how the Universe works.

Your work will be their heritage.

If we can't point at your work and say you did it we can't do anything else.

Code is speech

Daina,

I took the liberty of looking you up in the faculty directory. Thank you for looking into the code for the [redacted] computer program. The PI for the study was [redacted].

If you do find the code, I can arrange for it to be loaned to one of my colleagues at SAO.

Thank you, and if you need any more information from me, please let me know.

Hello,

At the end of the attached paper, there is a link to a computer code

[redacted]

The link does not work any more. Is it still possible to get the code?

Software Citation Principles

https://doi.org/10.7717/peerj-cs.86

- **Importance:** Software should be considered a legitimate and citable product of research...
- **Credit and attribution:** ...normative, legal attribution to all contributors to the software...
- **Unique identification:** ...identification that is machine actionable, globally unique, interoperable, and recognized...
- **Persistence:** Unique identifiers and metadata describing the software and its disposition should persist...
- **Accessibility:** ...access to the software itself and to its associated metadata...
- **Specificity:** ...identification of, and access to, the specific version of software...

FORCE11:

Software Citation Implementation Working Group

(implementing the software citation principles)

Recommendations Moving Forward

software metadata specifications:

- Citation File Format
- CodeMeta
- Software Heritage (universal software archive)

Citation File Format

human- and machine-readable file format that provides citation metadata for software.

Example

If you want to make your software easily citable, you can put a file called `CITATION.cff` in the root of your repository. This file should provide at least the minimally necessary metadata to cite your software. For example:

```yaml
cff-version: 1.0.3
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 Tool
version: 1.0.4
doi: 10.5281/zenodo.1234
date-released: 2017-12-18
```

Comprehensive Metadata for Software

- credit for academic software
  - citation metadata
- replicate some analysis
  - versions and dependencies
- discover software you don't already know
  - keywords and descriptions

Zenodo.org is a data archive based at CERN which is popularly used to archive and provide DOIs to academic software from GitHub, as described in the official GitHub guide to Making your code citable.

Property	Zenodo
codeRepository	relatedLink
applicationCategory	communities
author	creators
datePublished	date_published
funder	contributors.Funder
keywords	keywords
license	license
description	description/notes
identifier	id
name	title
affiliation	affiliation
identifier	ORCID
name	name

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

An essential infrastructure for science

Home / Mission / An essential infrastructure for science

A large part of the technical and scientific **knowledge** that is being developed today **resides in software**. The preservation of this universal body of knowledge has become as essential as preserving research articles and data sets.

As an extremely valuable **service to the research community**, we will search for, collect, organize, preserve and make easily available all the software.

Communities should build further guidance and develop **norms specific for their communities and use cases.**

Domain-Independent Terminology

Software is:

- **available as...**
  - open or closed source
- **archived when...**
  - preserved long term by someone other than the author
- **stewarded when...**
  - "actively" preserved
- **identified using...**
  - an identifier separate from the code's location (not a URL)
- **indexed when...**
  - the identifier is on a list maintained by an indexing service

"Published Software"

published = permanently archiving it and creating a resolvable identifier (e.g. by Zenodo, figshare, institutional archival repositories)

unpublished = the software is made available by a hosting organization that does not commit to long term preservation (e.g. GitHub, personal website)

Best thing to cite

"published software"

means

"archived software"

oh god

Case Study

(slightly better than anecdotes)

- recommend comprehensive reviewer and editor guidelines/inform policy development among publishers
- develop training and other resources for article authors and software developers to improve software citation practices
- inform the prioritization and development of tools that could be used to support software citation

WIP on GitHub

Sample of software packages developed in whole or in part at the CfA

Likely to be cited

Cover long year range

AAS XML (1998-2018)

ADS API search (forthcoming)

1. AstroBlend
2. AstroPy
3. SAOImage DS9
4. PlasmaPy
5. RADMC-3D
6. spec2d
7. Stingray
8. Tardis
9. WCSTools

Methods

Define "aliases" for the software packages.

Find out where aliases are found in articles.

- Reference = XML tag is a citation tag or the string associated with the alias contains bibtex record indicator
- Acknowledgement = XML tag
- Footnote = XML tag is a footnote tag or the string associated with the alias contains a known footnote indicator

How many articles contained some kind of "alias"?

AstroBlend: 2

AstroPy: 1168

RADMC-3D: 613

Spec2d: 610

Stingray: 9

TARDIS: 19

WCS Tools: 191

References?

AstroBlend: 0%

AstroPy: 86%

RADMC-3D: 68%

Spec2d: 71%

Stingray: 55%

TARDIS: 84%

WCS Tools: 54%

How often were identifiers used to give credit?

AstroBlend: 0%

AstroPy: 76%

RADMC-3D: 2.7%

Spec2d: 45%

Stingray: 0%

TARDIS: 21%

WCS Tools: 19.8%

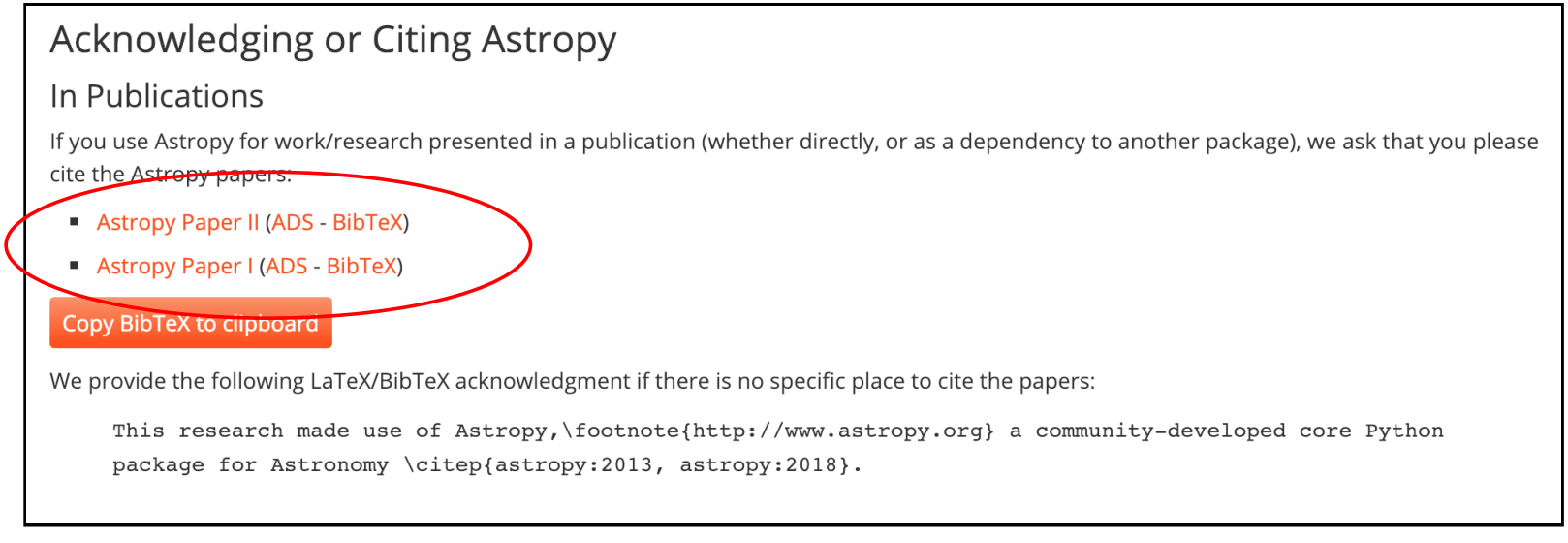
How often was a Software DOI used?

0%

Why?

Authors are specifically requesting people cite something other than the code even when a Zenodo DOI for the code exists.

These things should be cited in addition to the code, rather than as stand-ins for the code



Acknowledging or Citing Astropy

In Publications

If you use Astropy for work/research presented in a publication (whether directly, or as a dependency to another package), we ask that you please cite the Astropy papers:

- Astropy Paper II (ADS - BibTeX)
- Astropy Paper I (ADS - BibTeX)

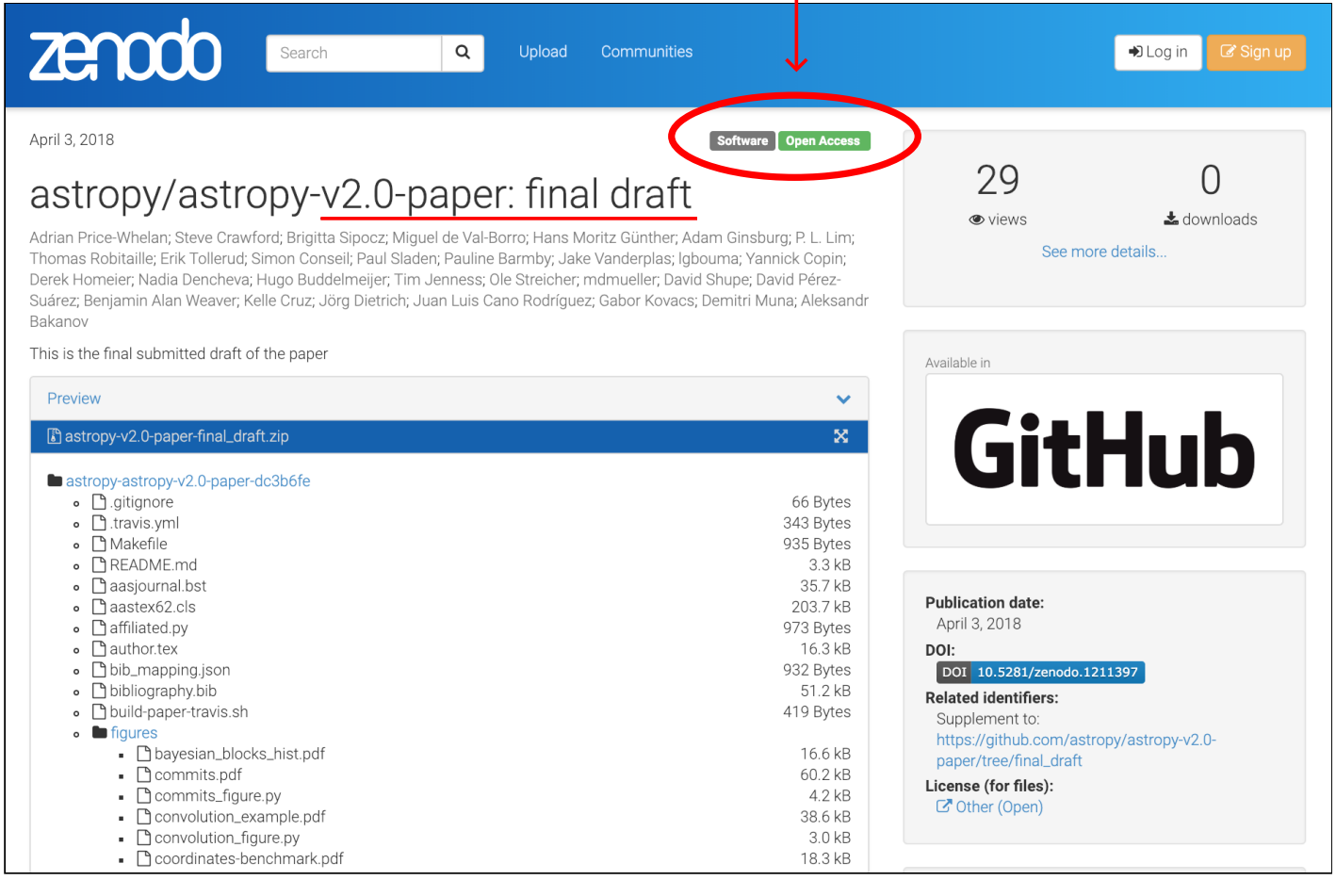
Copy BibTeX to clipboard

We provide the following LaTeX/BibTeX acknowledgment if there is no specific place to cite the papers:

```
This research made use of Astropy,\footnote{http://www.astropy.org} a community-developed core Python
package for Astronomy \citep{astropy:2013, astropy:2018}.
```

Software DOI doesn't guarantee a native software citation

```
<resourceType resourceTypeGeneral="Software"/>
```



PlasmaPy: an open source community-developed Python package for plasma physics

PlasmaPy Community; Murphy, Nicholas A.; Leonard, Andrew J.; Stańczak, Dominik; Kozlowski, Pawel M.; Langendorf, Samuel J.; Haggerty, Colby C.; Beckers, Jasper P.; Mumford, Stuart J.; Parashar, Tulasi N.; Huang, Yi-Min

BibTeX Export

```
@misc{plasmapy_community_2018_1238132,
  author       = {PlasmaPy Community and
                  Murphy, Nicholas A. and
                  Leonard, Andrew J. and
                  Stańczak, Dominik and
                  Kozlowski, Pawel M. and
                  Langendorf, Samuel J. and
                  Haggerty, Colby C. and
                  Beckers, Jasper P. and
                  Mumford, Stuart J. and
                  Parashar, Tulasi N. and
                  Huang, Yi-Min},
  title        = {{PlasmaPy: an open source community-developed
                   Python package for plasma physics}},
  month        = apr,
  year         = 2018,
  note         = {{This work was partially supported by the U.S.
                   Department of Energy.}},
  doi          = {10.5281/zenodo.1238132},
  url          = {https://doi.org/10.5281/zenodo.1238132}
}
```

Slide Deck

ASCL registry records are cited instead and often contain complicated or conflicting instructions

ASCL Code Record

[ascl:1109.015] WCSTools: Image Astrometry Toolkit

Mink, Jessica

WCSTools is a package of programs and a library of utility subroutines for setting and using the world coordinate systems (WCS) in the headers of the most common astronomical image formats, FITS and IRAF .imh, to relate image pixels to sky coordinates. In addition to dealing with image WCS information, WCSTools has extensive catalog search, image header manipulation, and coordinate and time conversion tasks. This software is all written in very portable C, so it should compile and run on any computer with a C compiler.

Code site: http://tdc-www.harvard.edu/software/wcstools/
Appears in: http://adsabs.harvard.edu/abs/1999ASPC..172..498M

Bibcode: 2011ascl.soft09015M

Preferred citation method:
Depends on usage; see http://tdc-www.harvard.edu/software/wcstools/publications/ for information

References

If the WCSTools package has proven useful in your work, please reference at least one of the following papers. If you want to find out which is most applic

Entire Package

As of 2018, the latest summary paper for WCSTools is being presented at ADASS XXVIII in College Park, MD and is entitled:
"Exploring Space, Time, and Data with WCSTools " and will be published in 2019 in Astronomical Data Analysis Software and Systems XXVIII
[abstract] [PDF poster] [HTML of the mostly-text poster]

In 2005, WCSTools was upgraded to the then-latest version of WCSLIB and a paper was presented at ADASS:
"WCSTools 4.0: Building Astrometry and Catalogs into Pipelines," Douglas J. Mink, 2005 in Astronomical Data Analysis Software and Systems XV,
[abstract] [PDF poster] [HTML presentation [PDF presentation] [PDF paper] [HTML paper]

The 2001 ADASS paper gives the current status of the package, with examples of many of its capabilities:
"WCSTools 3.0: More Tools for Image Astrometry and Catalog Searching", Douglas J. Mink, 2002, in Astronomical Data Analysis Software and Systems XI,
[abstract] [full text] [PDF poster]

The 1998 ADASS paper is the first published description of the tools in the WCSTools package, including SAOimage when used for WCS work:
"WCSTools: An Image Astrometry Toolkit", Douglas J. Mink (1998), in Astronomical Data Analysis Software and Systems VIII, A.S.P. Conference Series, V
[ADS abstract]

Fitting World Coordinate Systems to Images

The paper presented at ADASS in 1996 is the best published description of the IMWCS world coordinate system fitting program:
"WCSTools: Putting Image World Coordinate Systems to Use", Douglas J. Mink, 1997, in Astronomical Data Analysis Software and Systems VI, A.S.P. Conf
[ADS Abstract] [local abstract] [proceedings full text]

Using Source Catalogs

Our 2003 ADASS paper tests the accuracy of the various catalogs one is likely to use with **imwcs**, showing how well it works in a pipeline mode on 1728 ima
"A Comparison of Large All-Sky Catalogs" Douglas J. Mink, Warren R. Brown, and Michael J. Kurtz in Astronomical Data Analysis Software and Systems X
p. 141.
[abstract] [PDF full text] [Web presentation]

Our 2003 AAS/DDA paper has more of an astrometric slant comparing accuracies of the most commonly used catalogs against actual star positions in 1728 in
"A Comparison of Large All-Sky Catalogs", Douglas J. Mink, Warren Brown, and Michael J. Kurtz, 2003.
[abstract] [Web presentation]

The 2002 ADASS paper displays some of the uses of **scat** as part of the Virtual Observatory:
"Federating Catalogs and Interfacing Them with Archives: A VO Prototype" Douglas J. Mink and Michael J. Kurtz, 2002, in Astronomical Data Analysis Soft
ISSN: 1080-7926, 2002., p. 169.
[abstract] [PDF full text] [Web presentation] [Open/StarOffice presentation]

Metadata is scattered and not formatted uniformly

software authors

- You control your metadata.
- You are your own cataloger.

article authors

- You need to cite software correctly.
- No one else will catch mistakes.
- You are your own copy editor.

Publishers

- You need policies that can be enforced.
- You need to provide examples.

Things you can do

right now

Software Authors

- Mint a DataCite software DOI
- Create a CFF file
- License your data and code explicitly
- **Update and check your metadata**
  - Check it again
- Link documentation to the source code directly
- Ensure your preferred citations/any instructions about attribution **enable native software citation**
- If you have many versions of software, decide who the authors are for the "concept" of the software*
- Try out the new AAS/JOSS code review process

* get a freaking ORCiD

Article Authors

- Look for preferred citations
  - Look everywhere
- If you cannot find a preferred citation, follow the F11 guidance and make sure you're doing your best at native software citation
- Consider the version that you are citing
  - Who are you trying to give credit?
- Follow publisher policies, if there isn't one follow F11 principles
- **Put software citations in the references section**
- **Cite your own code in a software paper**
  - tells others how you want it cited

Publishers

- Make a software citation policy
- Provide examples
  - What to do
  - What not to do
- Make expectations clear as to how much editorial review will be dedicated to checking software citations
  - **Everyone assumes you will fix it**
- If you accept software papers recommend authors create metadata files and mint a DOI
  - Provide examples of these

Thank You