

Setting the Scene to Cite Software

Jez Cope, British Library

Neil Chue Hong, Software Sustainability Institute

Frances Madden, British Library



Why cite software?

1. Reproducibility

2. Credit

3. Reuse

Citation isn't the only solution, but it is an interesting one

Citation for reproducibility

- Trust in results requires reproducibility
- Page limits
+ modern analysis
= incomplete picture
- Software code specifies the method precisely

1²/₃ C.(10 oz.) dry black be
2 bay leaves, 1t.minced garl
onion, $\frac{1}{2}$ t. dried oregano le
ground*~~onion~~ cumin, 1 $\frac{1}{2}$ t.sal
2 T. wine vinegar, $\frac{1}{4}$ C.choppe
 $\frac{1}{4}$ C.chopped onion.*****
Place beans & water in a hea
cover & let soak in cool plac
cook, add bay leaves to beans
to boil over moderate heat;
low & simmer 1 hr. Remove pan
garlic, the 2/3 C.onion, oregan
r.Return pan to heat & simmer
checking every 30 min. & addi
ove:

Citation for credit

- Citations are (still) the coin of the realm
- RSEs not always named on papers
- Citing software gives credit to *software* authors



Citation for reuse

- A paper with easy access to the code is:
 - ...easier to build new research on...
 - ...thus more likely to be cited itself



How is software cited in the scholarly literature?

TABLE 6. Types of software mentions in publications.

Mention type	Count (<i>n</i> = 286)	Proportion (95% CI)	Example
Cite to publication	105	0.37 (0.31–0.43)	<i>... was calculated using biosys (Swofford & Selander 1981).</i>
Cite to users manual	6	0.02 (0.01–0.05)	<i>... as analyzed by the BIAevaluation software (Biacore, 1997).</i> Reference List has: Biacore, I. (1997). BIAevaluation Software Handbook, version 3.0 (Uppsala, Sweden: Biacore, Inc)
Cite to project name or website	15	0.05 (0.03–0.09)	<i>... using the program Autodecay version 4.0.29 PPC (Eriksson 1998).</i> Reference List has: ERIKSSON, T. 1998. Autodecay, vers. 4.0.29 Stockholm: Department of Botany.
Instrument-like	53	0.19 (0.14–0.24)	<i>... calculated by t-test using the Prism 3.0 software (GraphPad Software, San Diego, CA, USA).</i>
URL in text	13	0.05 (0.03–0.08)	<i>... freely available from http://www.cibiv.at/software/pda/.</i>
In-text name mention only	90	0.31 (0.26–0.37)	<i>... were analyzed using MapQTL (4.0) software.</i>
Not even name mentioned	4	0.01 (0.00–0.04)	<i>... was carried out using software implemented in the Java programming language.</i>

Examples from Biology Literature

Howison and Bullard, <https://doi.org/10.1002/asi.23538>

SOFTWARE CITATION PRINCIPLES

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; they should be included in the metadata of the citing work, such as a reference list. Software should be cited on the same basis as any other research product such as a paper or a book.

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.

PERSISTENCE

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

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.

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.

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.

Challenges to updating the scholarly ecosystem

- Software not indexed in bibliometric systems
- Problems with Citation Style Language (CSL) and BibTeX
 - No software type (treated as book/misc)
- Ensure metadata not lost in transit (JATS4R, CodeMeta, Schema.org)
- Software “concept” vs “version”
 - Aggregating citations
- “Publisher” concept doesn’t directly map
- Citing and identifying closed source or unpublished software
- Identifying software used within research process, not just identifying software in a paper

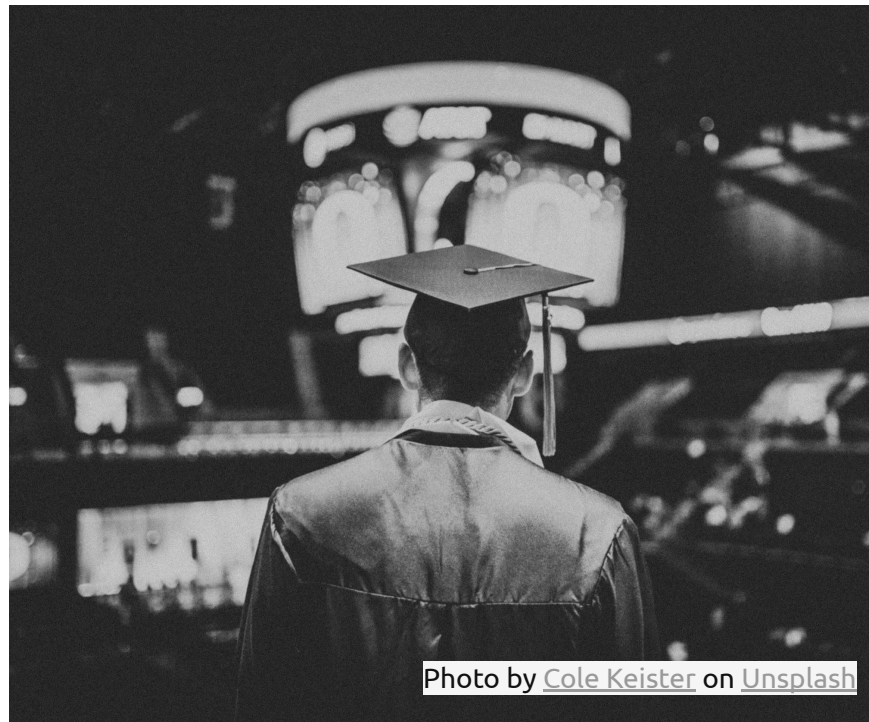


Photo by [Cole Keister](#) on [Unsplash](#)

Roles in improving software citation

Researcher: *cite software properly, make work reproducible*

RSE: *provide recommended citation, deposit software to get persistent identifier*

Publisher: *provide guidance on software citation*

Repositories: *publish software, show impact*

Library / Archive: *provide guidance, services*

Funder / Policymaker: *create and adopt policies, show impact of software*

Infrastructure Providers: *include software in indexes/catalogues/tools*



Photo by [Andrew Seaman](#) on [Unsplash](#)

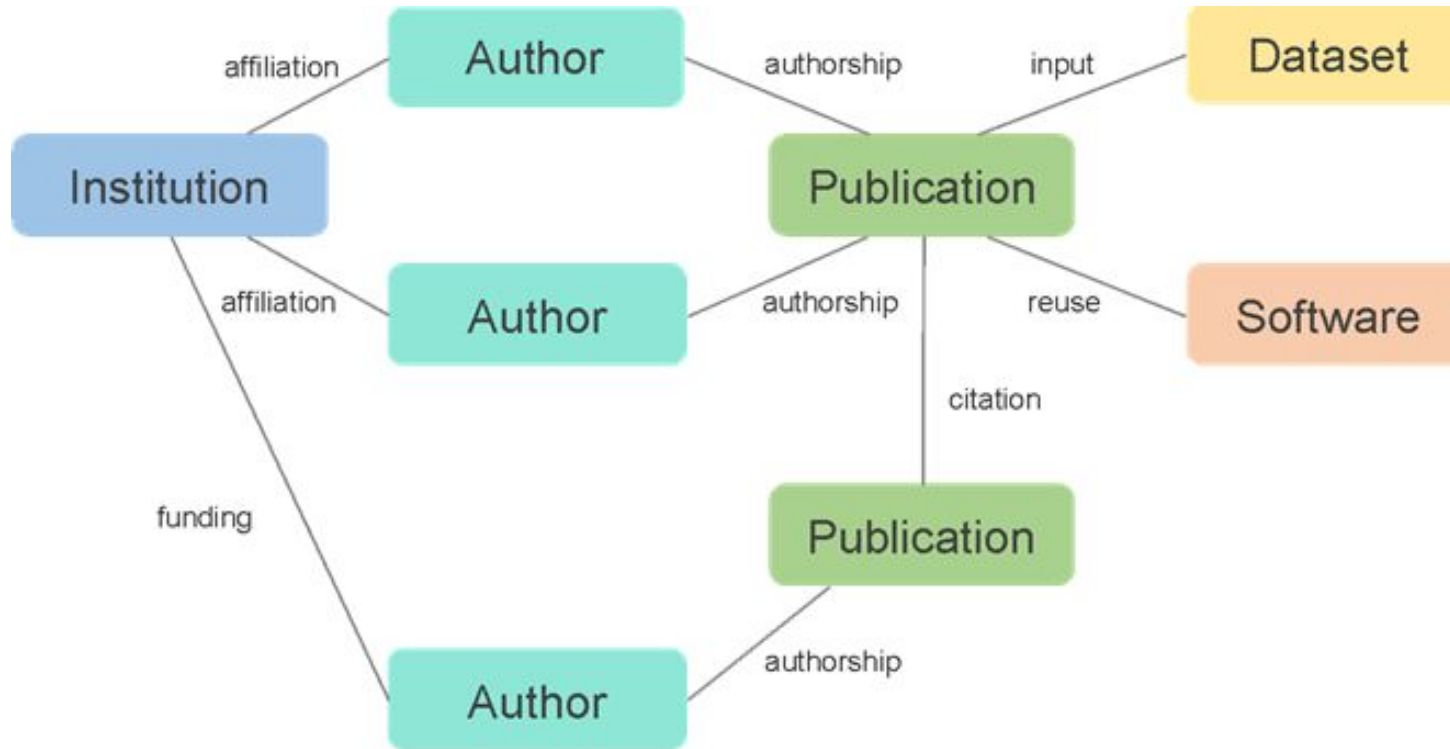


FREYA

- Connected Open Identifiers for Discovery, Access and Use of Research Resources
- “... iteratively extend a robust environment for Persistent Identifiers (PIDs) into a core component of European and global research e-infrastructures”
- Interested in building the infrastructure around software to connect it in with the rest of the scholarly record
- Need to be able to track software so it does not get lost
- PIDs can help us do this!



PID Graph



Software Citation Requirements Brainstorming

- Provide three things that would **enable** your “ideal” software citation scenario
 - Where do you want to get to?
 - What would the ideal scenario look like?
- Write them down on the Post-It notes on your table
- Cluster the Post-It notes with other similar requirements

Breakout Groups 1: Software Citation Challenges

- Nominate a scribe/rapporteur
- What challenges are there in the way of meeting the requirements identified before lunch? What is stopping us from reaching our “ideal scenario”?
- How would we prioritise these challenges by urgency, importance and feasibility
- *Come up with three challenges to report back*

Breakout Groups 2: Software Citation Solutions

- Nominate a scribe/rapporteur
- Discuss solutions to address the challenges
- Choose one solution for your group to focus on
- Describe how this solution would work and what challenges it addresses.
 - How much funding would it require, who would be the funders, which people / organisations should be involved?
- *Report back on your solution*