PeerJ Computer Science Symposium: Software Citation: Principles, Guidance, Challenges, and Progress

8 December 2022

Neil P. Chue Hong, Software Sustainability Institute/University of Edinburgh
Daniel S. Katz, University of Illinois at Urbana-Champaign
N.ChueHong@software.ac.uk, d.katz@ieee.org

https://doi.org/10.5281/zenodo.7409765

Poll:
What’s your opinion about software citation?

- Skeptical
- Interested
- Motivated
- Working on
- Unsure

Introduction

The practice of software citation has progressed substantially over the last decade, but there are still challenges. This PeerJ special issue gives a chance to reflect on the status of research investigating the importance of - and best practice for - citing, indexing, and discovering software used as a scholarly research tool.

While the ideas around software citation have been discussed for many years, recent work has coalesced around the FORCE11 Working Group to develop best practices for authors and journals for software citation, working with others in the community.

FORCE11 group products includes:

- Software citation implementation challenges
- Checklists for (paper) authors and (software) developers
- Best practices for software repositories and registries
- Guidance for journals

https://doi.org/10.5281/zenodo.7409765

The FORCE11 Software Citation journey

Software Citation WG started

~55 members (researchers, developers, publishers, repositories, librarians)

Reviewed existing community practices & developed use cases

Software Citation Principles published

Started with data citation principles, updated based on software use cases and related work, working group discussions, community feedback

Software citation principles published after community review: 10.7717/peerj-cs.86

https://doi.org/10.5281/zenodo.7409765

FORCE11 Software Citation Working Group (2015-16)

- Documented differences between software and data; defined software citation challenges
  - Katz DS, Niemeyer KE, et al. (2016) Software vs. data in the context of citation. PeerJ Preprints 4:e2630v1. DOI: 10.7287/peerj.preprints.2630v1
  - Niemeyer KE, Smith AM, Katz DS. (2016) The challenge and promise of software citation for credit, identification, discovery, and reuse. ACM Journal of Data and Information Quality, 7(4):16. DOI: 10.1145/2968452
- Created software citation principles
  - Smith AM, Katz DS, Niemeyer KE, FORCE11 Software Citation Working Group. (2016) Software Citation Principles. PeerJ Computer Science 2:e86. DOI: 10.7717/peerj-cs.86 and https://www.force11.org/software-citation-principles

https://www.force11.org/group/software-citation-working-group
Co-Chairs: Arfon M. Smith, Daniel S. Katz, Kyle E. Niemeyer

Importance of Software Citation

- Software is as integral as a paper or dataset for facilitating full understanding and dissemination of research
  - Books and journal articles have long benefited from an infrastructure that makes them easy to cite, a key element in the process of research and academic discourse in all disciplines
  - Software (e.g., code, scripts, models, notebooks and libraries) should be cited in the same way

- Citing software helps further research & provides a means for other researchers to access software in order to
  - support proper attribution and credit (similar to that of papers, data, etc.)
  - enable peer-review, validation, and reproducibility of findings
  - support collaboration and reuse
  - encourage building on the work of others

- Software citation elevates software to the level of a first-class object in the digital scholarly ecosystem, consistent with its immense actual present-day significance

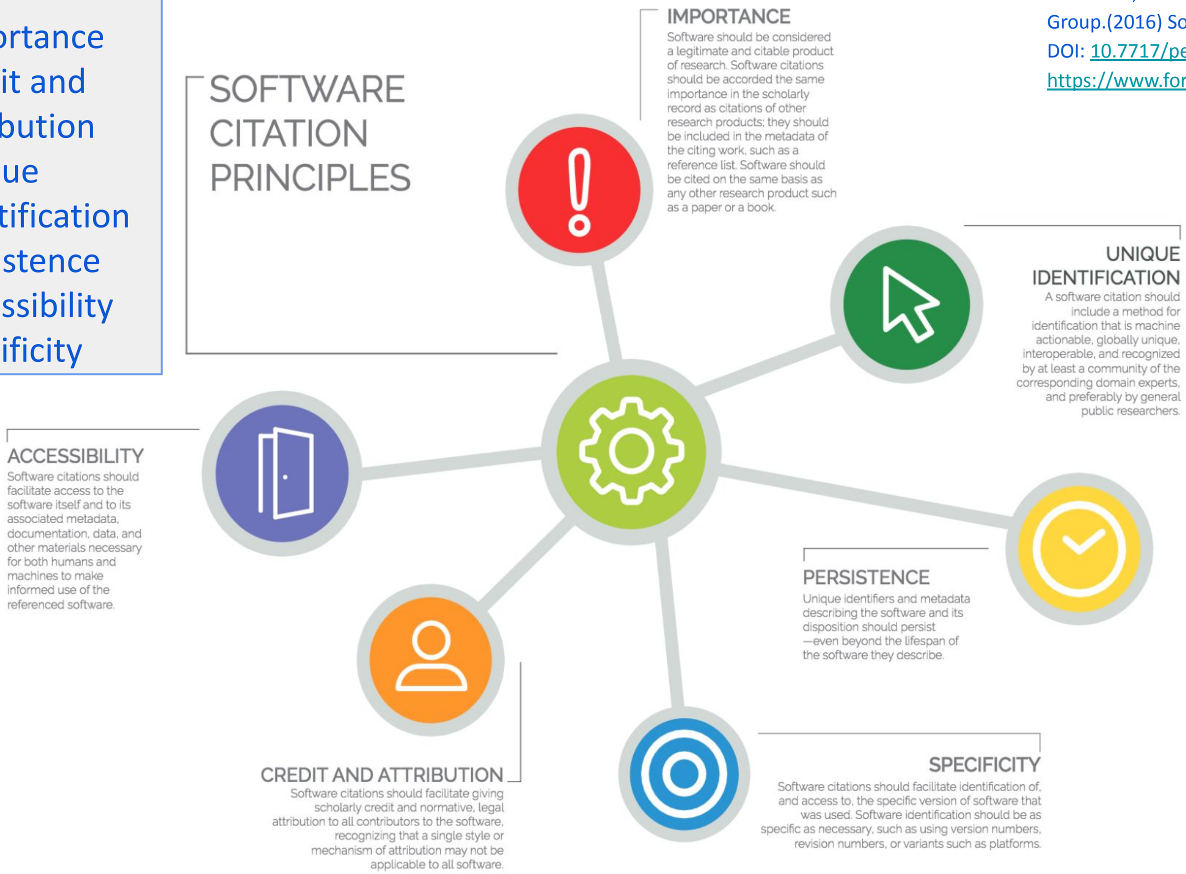
- More information on the current state of software citation later…

https://doi.org/10.5281/zenodo.7409765

Software is a critical part of modern research...

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

Smith AM, Katz DS, Niemeyer KE, FORCE11 Software Citation Working Group.(2016) Software Citation Principles. PeerJ Computer Science 2:e86. DOI: 10.7717/peerj-cs.86 and https://www.force11.org/software-citation-principles



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.

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.

PERSISTENCE

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

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.

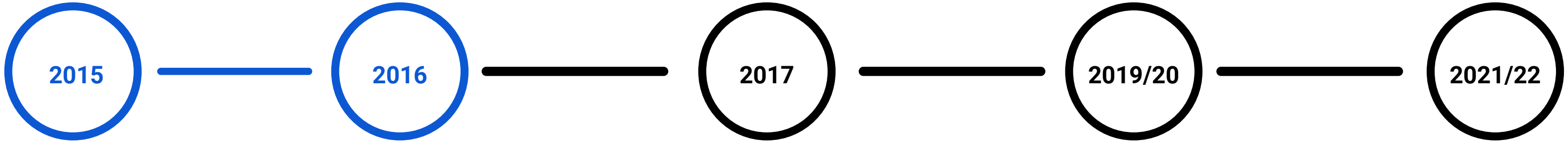
SPECIFICITY

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.

Image courtesy of DataCite

... yet there is little support for its acknowledgement and citation

The FORCE11 Software Citation journey



2015

Software Citation WG started

~55 members (researchers, developers, publishers, repositories, librarians)

Reviewed existing community practices & developed use cases

2016

Software Citation Principles published

Started with data citation principles, updated based on software use cases and related work, working group discussions, community feedback

Software citation principles published after community review: 10.7717/peerj-cs.86

2017

Software Citation Implementation WG started

Group set up to:
1. endorse the principles
2. develop sets of guidelines for implementing the principles
3. help implement the principles
4. test specific implementations of the principles

2019/20

Task forces publish resources

Guidance Task Force develops checklists for paper authors: 10.5281/zenodo.3479198 and software developers: 10.5281/zenodo.3482768

CodeMeta Task Force provides recommendations for schema changes

Repositories Task Force publishes best practices: https://arxiv.org/abs/2012.13117

Journals Task Force publishes "Recognizing the value of software: a software citation guide" 10.12688/f1000research.26932.2

2021/22

Adoption of software citation increases?

Journals Task Force starts adoption process with journals and publishers

Examine and Improve how citations are processed and handed off to indexers (Crossref, DataCite)

FORCE11
The Future of Research Communications and e-Scholarship

https://doi.org/10.5281/zenodo.7409765

FORCE11 Software Citation Implementation Working Group (2017-present)

- Initial goals:
  - Write out the “small amount” of detail needed to implement the principles
  - Coordinate research & other work going on in many areas
  - Work with communities to actually implement the principles
- Quickly realized “small amount” of detail wasn’t small, scattered progress wasn't sufficient, underlying challenges not being addressed
  - D. S. Katz, et al., "Software Citation Implementation Challenges", arXiv 1905.08674 [cs.CY], 2019.

https://www.force11.org/group/software-citation-implementation-working-group

Co-Chairs: Neil Chue Hong, Martin Fenner, Daniel S. Katz

Software Citation Challenges (2019)

- Technical
  - Complexity of software types: open source, closed source; published, unpublished; versioned, unversioned; developed by citer, not developed by citer; services, containers, executables
  - How to uniquely identify software of each type (ideally as uniformly as possible)
    - Worked on by joint FORCE11/RDA Software Source Code Identification WG
  - How to define and store citation metadata for each type
  - How to access metadata and convert it as needed
  - How to count citations across versions
  - Realization: metadata is fundamental
- Social
  - Need groups that work on implementation in context: disciplinary communities, publishers, repositories, indexers, funders, institutions
  - Groups need to come together, run pilots to establish norms

https://doi.org/10.5281/zenodo.7409765

Responses to challenges (1)

- Guidance task force
  - For paper authors who want to cite software
    - N. P. Chue Hong, et al., "Software Citation Checklist for Authors," Zenodo, 15-Oct-2019. https://doi.org/10.5281/zenodo.3479198
  - For software developers who want to make their software citable
    - N. P. Chue Hong, et al., "Software Citation Checklist for Developers," Zenodo, 15-Oct-2019. https://doi.org/10.5281/zenodo.3482768
- CodeMeta task force
  - Following CodeMeta project
    - Aiming to understand metadata for software, not just for use in citation
    - Built a crosswalk of existing metadata standards for software
    - Then developed a CodeMeta standard to describe software based on these crosswalks
  - Updating the CodeMeta standard
  - Describing everything in CodeMeta using schema.org properties
  - Moving CodeMeta into a community group, with governance

https://doi.org/10.5281/zenodo.7409765

Responses to challenges (2)

- Software Registries Task Force
  - Developed best practices document
    - D. Garijo et al., "Nine Best Practices for Research Software Registries and Repositories," 2022. PeerJ CS 8:e1023, https://doi.org/10.7717/peerj-cs.1023
  - Community continuing in SciCodes: Consortium of scientific software registries and repositories, https://scicodes.net/
- Journals Task Force
  - Working with publishers to provide generic guidelines for journals and conferences to provide to authors
    - They then provide specific guidelines, with community-accepted language and examples
    - D. S. Katz, et al., "Recognizing the value of software: a software citation guide [version 2; peer review: 2 approved]," F1000Research 9:1257, 2021. https://doi.org/10.12688/f1000research.26932.2
    - Tracked by CHORUS in Software Citation Policy Index
  - Also working on publication processing
    - How citation information moves from author provides to internal publisher/contractor systems and then to indices
    - S. Stall, et al., "Journal Production Guidance for Data and Software Citations", in draft

Responses to challenges (3)

- Considered institutions task force
  - Institutions: places where people work
    - Universities, laboratories, industry, government, etc.
  - Want to affect policies and practices
    - How do they encourage software citation
    - How do they use software citation information in hiring & promotion
  - Collect and share examples
  - Help form communities
  - But insufficient interest from FORCE11 WG members
- Given progress to date, what else makes sense to do, and who can do it?
  - Recent IMLS-funded software citation workshop (led by Daina Bouquin) to assess and plan next steps
    - Report coming soon

https://doi.org/10.5281/zenodo.7409765

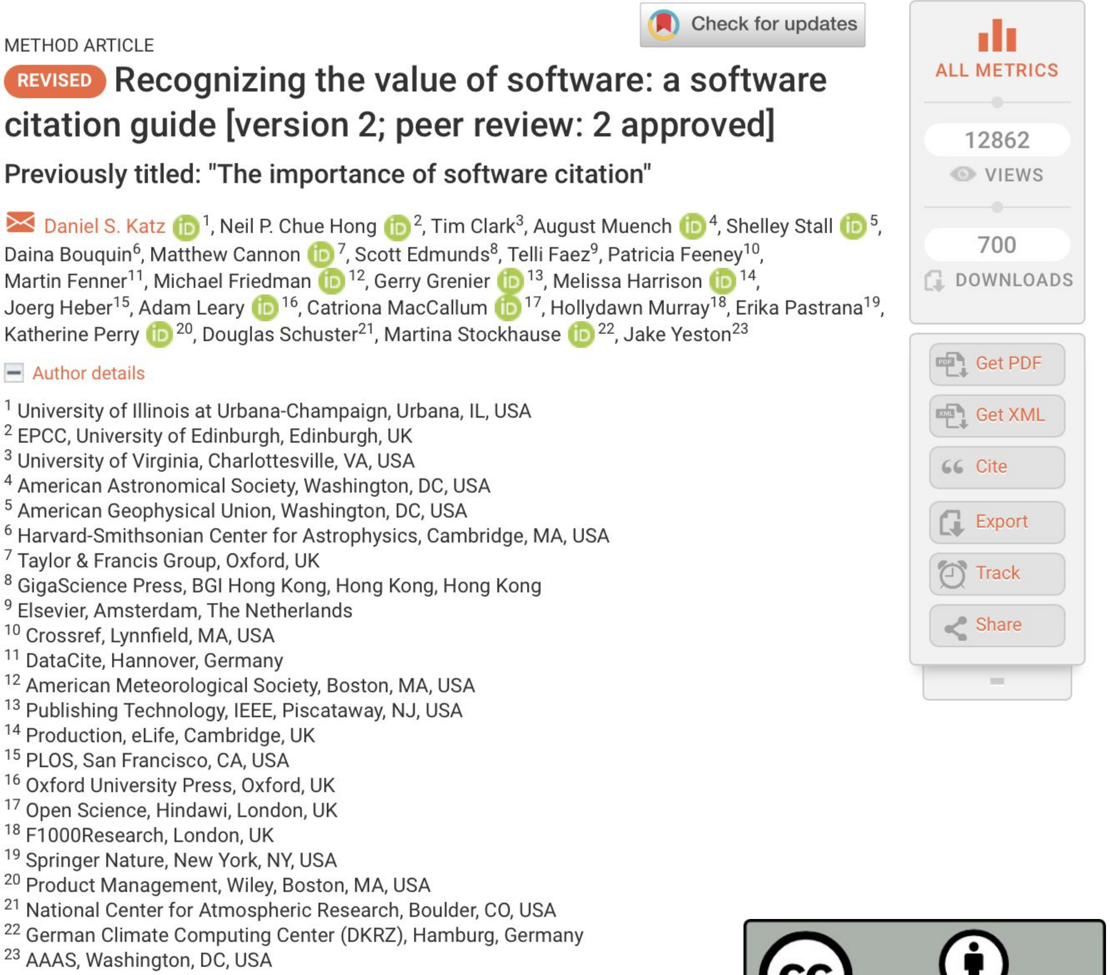
Poll:
Are you aware of journal policies that include software citation?

- Yes, and I use them
- Yes, but not currently using them
- No

https://doi.org/10.5281/zenodo.7409765

Journal Guidance

- In collaboration with publishers
  - American Astronomical Society, American Geophysical Union, Taylor & Francis, GigaScience, Elsevier, American Meteorological Society, IEEE, eLife, PLOS, Oxford University Press, Hindawi, F1000Research, Springer Nature, Wiley, AAAS
- And other stakeholders
  - Harvard-Smithsonian Center for Astrophysics, Crossref, DataCite, National Center for Atmospheric Research, German Climate Computing Center (DKRZ)
- Also intended to apply to conference proceedings



METHOD ARTICLE

REVISED **Recognizing the value of software: a software citation guide [version 2; peer review: 2 approved]**

Previously titled: "The importance of software citation"

Daniel S. Katz[1], Neil P. Chue Hong[2], Tim Clark[3], August Muench[4], Shelley Stall[5], Daina Bouquin[6], Matthew Cannon[7], Scott Edmunds[8], Telli Faez[9], Patricia Feeney[10], Martin Fenner[11], Michael Friedman[12], Gerry Grenier[13], Melissa Harrison[14], Joerg Heber[15], Adam Leary[16], Catriona MacCallum[17], Hollydawn Murray[18], Erika Pastrana[19], Katherine Perry[20], Douglas Schuster[21], Martina Stockhause[22], Jake Yeston[23]

Author details

[1] University of Illinois at Urbana-Champaign, Urbana, IL, USA
[2] EPCC, University of Edinburgh, Edinburgh, UK
[3] University of Virginia, Charlottesville, VA, USA
[4] American Astronomical Society, Washington, DC, USA
[5] American Geophysical Union, Washington, DC, USA
[6] Harvard-Smithsonian Center for Astrophysics, Cambridge, MA, USA
[7] Taylor & Francis Group, Oxford, UK
[8] GigaScience Press, BGI Hong Kong, Hong Kong, Hong Kong
[9] Elsevier, Amsterdam, The Netherlands
[10] Crossref, Lynnfield, MA, USA
[11] DataCite, Hannover, Germany
[12] American Meteorological Society, Boston, MA, USA
[13] Publishing Technology, IEEE, Piscataway, NJ, USA
[14] Production, eLife, Cambridge, UK
[15] PLOS, San Francisco, CA, USA
[16] Oxford University Press, Oxford, UK
[17] Open Science, Hindawi, London, UK
[18] F1000Research, London, UK
[19] Springer Nature, New York, NY, USA
[20] Product Management, Wiley, Boston, MA, USA
[21] National Center for Atmospheric Research, Boulder, CO, USA
[22] German Climate Computing Center (DKRZ), Hamburg, Germany
[23] AAAS, Washington, DC, USA

https://doi.org/10.5281/zenodo.7409765

Software citation - Recommended items

The following should be provided as part of the reference:

- Creator(s): the authors or project that developed the software.
- Title: the name of the software.
- Publication venue: the publication venue of the software, preferentially, an archive or repository that provides persistent identifiers.
- Date: the date the software was published. This is the date associated with a release or version of the software, or “n.d.” if the date is unknown.
- Identifier: a resolvable pointer to the software, preferentially, a PID that resolves to a landing page containing descriptive metadata about the software, similar to how a Digital Object Identifier (DOI) for a paper that points to a page about the paper rather than directly to a representation of the paper, such as the PDF. DOIs are preferable, and other examples of PIDs include Handles, RRIDs, ASCL IDs, swMath IDs, Software Heritage IDs, ARKs, etc. If there is no PID for the software, a URL to where the software exists may be the best identifier available.

https://doi.org/10.5281/zenodo.7409765

Software citation - Optional items & papers

Depending upon the publisher, the following may be required:

- Version: the identifier for the version of the software being referenced. If the version is unidentified or unknown, the date of access should be used.
- Type: some citation styles (e.g., APA), require a bracketed description of the citation (e.g., Computer software) to be included.

If an article exists that describes the software, it should be cited as an additional reference, as well as citing the software itself

- Do not cite the article instead of the software

https://doi.org/10.5281/zenodo.7409765

Software Citation Guidance Details

Related Work section:

“In the field of numerical software, a different approach is taken by BLAS (BLAS team, n.d).”

“Finite element methods are common in production codes, such as ConnectFlow® (Jacobs Engineering Group Inc., n.d.). because of their geometrical flexibility.”

Methodology section:

“In this paper, we evaluate integrated surface/subsurface permafrost thermal hydrology models implemented in the Advanced Terrestrial Simulator (ATS) v0.88 (Coon et al., 2020).”

“We then performed a PCA including the molar percentages of neutral sugars (Fuc, Rha, Ara, Gal, Glc, ManXyl) using the prcomp function in R version 3.4.2. (R Core Team, 2017)”

References

- BLAS team (n.d.), BLAS (Basic Linear Algebra Subprograms) [Computer software]. Netlib. http://www.netlib.org/blas/
- Coon, E., Berndt, M., Jan, A., Svyatsky, D., Atchley, A., Kikinzon, E., … Molins, S. (2020). Advanced Terrestrial Simulator (ATS) v0.88 (Version 0.88) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.3727209
- Jacobs Engineering Group Inc. (n.d.). ConnectFlow® [Computer software]. Jacobs Engineering Group Inc. https://www.jacobs.com/projects/ConnectFlow
- R Core Team. (2017). R: A language and environment for statistical computing (Version 3.4.2). Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.r-project.org

Notes: First is citation of a software concept, second for commercial software where only the executable is available, third an ideal citation following guidance with a long author list, fourth using a recommended citation for a core part of R software

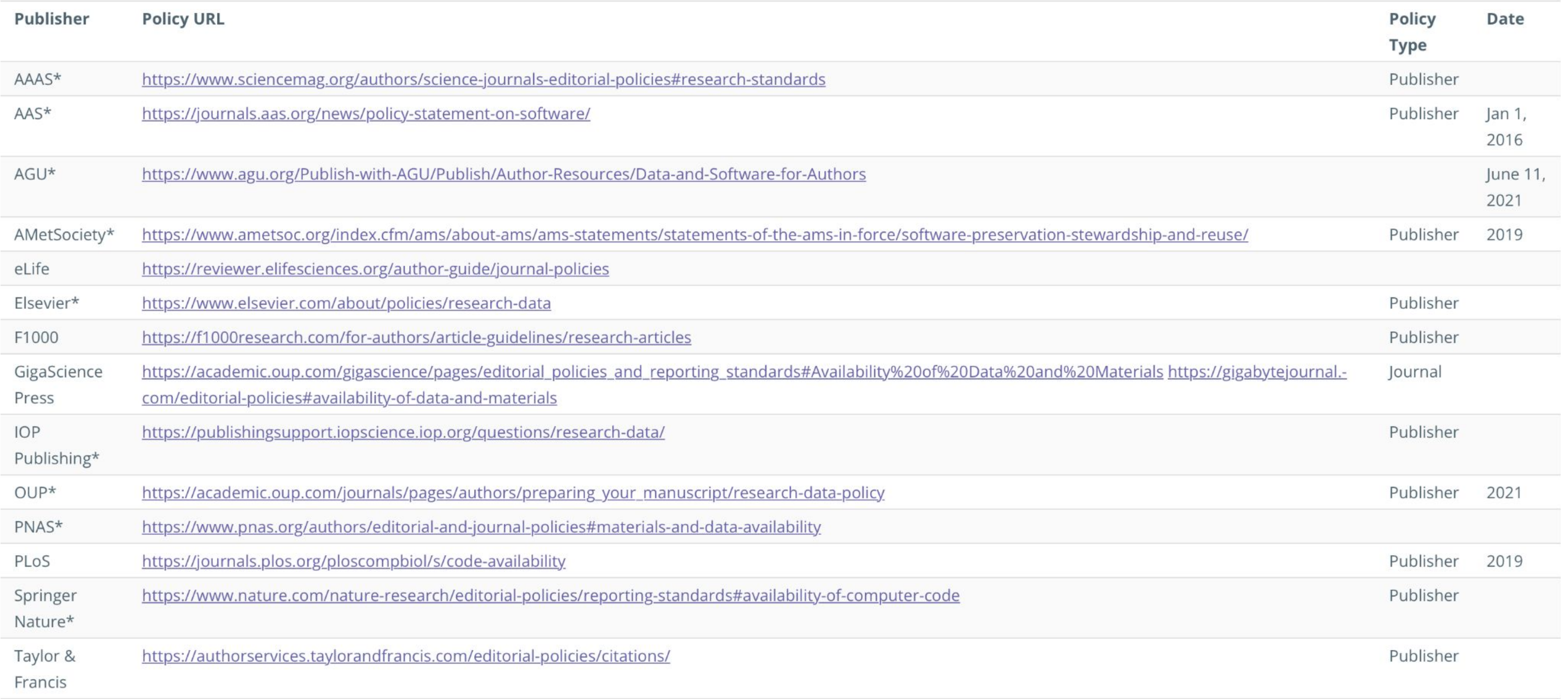
https://doi.org/10.5281/zenodo.7409765

Example policies: PeerJ Computer Science

- PeerJ Computer Science Code & Data Availability Policy
  - "All software needed to reproduce the results of the submission … must be made available"
  - "Third-party software must be cited appropriately, with a DOI (referred) or Software Heritage ID provided where available."
    - Link goes to paper: "Recognizing the value of software: a software citation guide [version 2; peer review: 2 approved]"
- PeerJ Computer Science Instructions to Authors:
  - "A DOI for software can be created by free services such as Zenodo. …"
  - "If making use of previously published author-written (or third-party) software, this must be cited appropriately, with a persistent identifier (a DOI or Software Heritage ID) provided where available. …"
  - "If making use of third-party code for which a persistent identifier does not exist, the software writer(s) MUST be acknowledged. Authors should make all reasonable attempts to ensure that the unpublished third-party software can be found by subsequent readers, and encourage the software writer to preserve their project by obtaining a persistent identifier."

https://doi.org/10.5281/zenodo.7409765

CHORUS Publisher Software Availability Policies Index



Publisher	Policy URL	Policy Type	Date
AAAS*	https://www.sciencemag.org/authors/science-journals-editorial-policies#research-standards	Publisher	
AAS*	https://journals.aas.org/news/policy-statement-on-software/	Publisher	Jan 1, 2016
AGU*	https://www.agu.org/Publish-with-AGU/Publish/Author-Resources/Data-and-Software-for-Authors		June 11, 2021
AMetSociety*	https://www.ametsoc.org/index.cfm/ams/about-ams/ams-statements/statements-of-the-ams-in-force/software-preservation-stewardship-and-reuse/	Publisher	2019
eLife	https://reviewer.elifesciences.org/author-guide/journal-policies		
Elsevier*	https://www.elsevier.com/about/policies/research-data	Publisher	
F1000	https://f1000research.com/for-authors/article-guidelines/research-articles	Publisher	
GigaScience Press	https://academic.oup.com/gigascience/pages/editorial_policies_and_reporting_standards#Availability%20of%20Data%20and%20Materials https://gigabytejournal.com/editorial-policies#availability-of-data-and-materials	Journal	
IOP Publishing*	https://publishingsupport.iopscience.iop.org/questions/research-data/	Publisher	
OUP*	https://academic.oup.com/journals/pages/authors/preparing_your_manuscript/research-data-policy	Publisher	2021
PNAS*	https://www.pnas.org/authors/editorial-and-journal-policies#materials-and-data-availability		
PLoS	https://journals.plos.org/ploscompbiol/s/code-availability	Publisher	2019
Springer Nature*	https://www.nature.com/nature-research/editorial-policies/reporting-standards#availability-of-computer-code	Publisher	
Taylor & Francis	https://authorservices.taylorandfrancis.com/editorial-policies/citations/	Publisher	

https://www.chorusaccess.org/resources/software-citation-policies-index/

https://doi.org/10.5281/zenodo.7409765

Software in the scientific literature (2016)

Citation to a related object

Difficult to assign credit

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

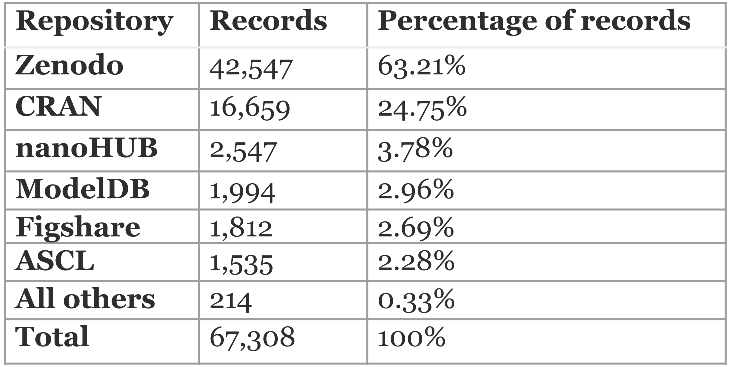
No PID

Table adapted from Table 6 in Howison and Bullard (2016). https://doi.org/10.1002/asi.23538

https://doi.org/10.5281/zenodo.7409765

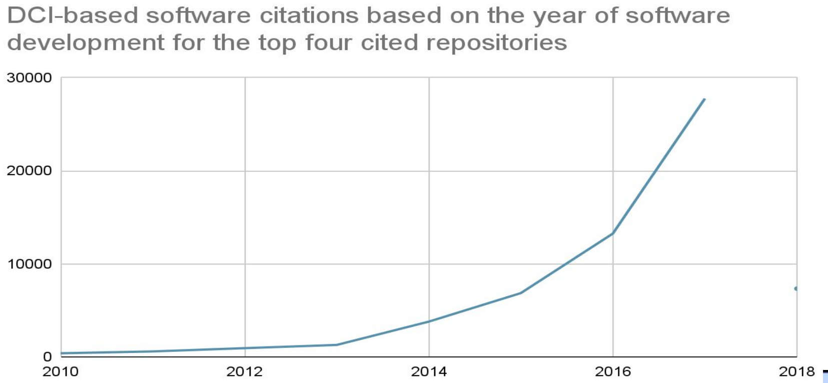
Software citation in the Data Citation Index (2019)

“Research software was rarely cited in the [Clarivate Analytics’ Data Citation Index], suggesting that the documented reuse of research software rarely occurs or is not well documented. Institutional repositories attracted few citations and had low rate of citation. It proved impossible, however, using the available data to isolate specific identifiers that can promote formal software citation.” Park and Wolfram (2019) https://doi.org/10.1016/j.joi.2019.03.005



Repository	Records	Percentage of records
Zenodo	42,547	63.21%
CRAN	16,659	24.75%
nanoHUB	2,547	3.78%
ModelDB	1,994	2.96%
Figshare	1,812	2.69%
ASCL	1,535	2.28%
All others	214	0.33%
Total	67,308	100%

Table 1. Relative distribution of software sharing records among the top repositories indexed by DCI.


DCI-based software citations based on the year of software development for the top four cited repositories

Adapted from Table 4. DCI-based software citations based on the year of software development for the top four cited repositories.

https://doi.org/10.5281/zenodo.7409765

Software mentions in the biomedical literature (2022)

“We disambiguate 1.12 million unique strings [from the NIH PMC-OA Commercial subset of 2.4 million paper] extracted by the NER model into 97,600 unique software entities, covering 78% of all software-paper links. We link 185,000 of the mentions to a repository, covering about 55% of all software-paper links.” Istrate et al. (2022)
https://doi.org/10.48550/arXiv.2209.00693 & https://doi.org/10.5061/dryad.6wwpzgn2c

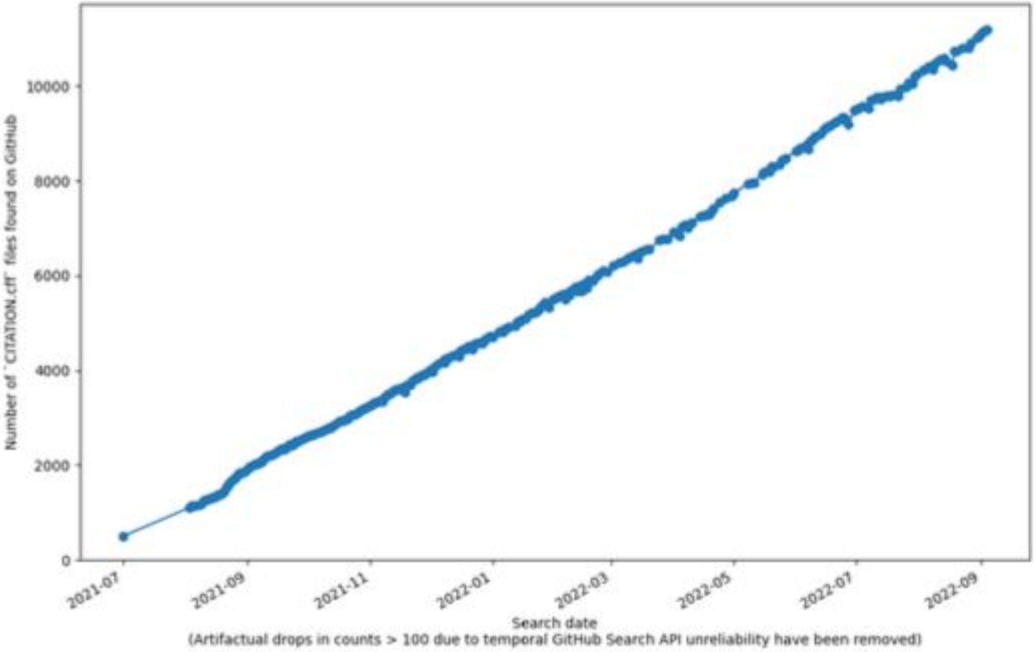
Abaqus, ABAQUS, ABC, ABySS, Adam, ADMIXTURE, Adobe, Adobe Illustrator, Adobe Photoshop, AFNI, Agilent Feature Extraction, AlexNet, Aligner, AlphaSim, AMBER, Amber, Amira, AMOS, Andromeda, ANNOVAR, ANSYS, antiSMASH, ARACNE, ARB, ArcGIS, ArcMap, Arlequin, ARLEQUIN, Artemis, ASTRAL, ATLAS.ti, Atlas.ti, AUGUSTUS, Augustus, AutoDock, Autodock, Autodock Vina, AutoDock Vina, AxioVision, Axiovision, Basic Local Alignment Search Tool (BLAST), BayeScan, bcftools, BD FACSDiva, Beagle, BEAGLE, BEAST, BEDTools, bedtools, BEST, BestKeeper, BiNGO, Bioconductor, BioEdit, Bioedit, BioMart, BioNJ, BioNumerics, BioRender, Bismark, Blast, blast, BLAST, BLAST+, BLAST2GO, Blast2GO, BLASTN, BLASTn, BlastN, Blastn, blastn, BLASTP, BLASTp, blastp, BlastP, Blastp, blastx, BLASTX, BLASTx, BlastX, Blastx, BLAT, Blender, bowtie, Bowtie, Bowtie2, bowtie2, Burrows, BUSCO, BWA, BWA-MEM, CADD, CalcuSyn, Canu, CAP3, car, CART, CASAVA, cBioPortal, CCP4, CD-HIT, CDOCKER, Cell Quest, CellML, CellProfiler, CellQuest, CellQuest Pro, CFX Manager, CHARMM, CheckM, Chimera, ChromHMM, CIBERSORT, Circos, CiteSpace, Clampfit, CLC Genomics Workbench, ClueGO, ClusPro, Clustal, Clustal Omega, Clustal W, CLUSTAL W, Clustal X, ClustalW, CLUSTALW, ClustalW2, ClustalX, Cluster, clusterProfiler, CNN, COBRA, CODEML, COLLECT, ComBat, Comprehensive Meta-Analysis, CompuSyn, COMSOL, COMSOL Multiphysics, ConSurf, Coot, COOT, Covidence, CrysAlis PRO, CrystalClear, CrystalStructure, CUDA, Cuffdiff, Cufflinks, cutadapt, Cutadapt, cytoHubba, Cytoscape, DADA2, DALI, DARTEL, Database for Annotation, Visualization and Integrated Discovery (DAVID), DAVID, DBSCAN, dChip, DENZO, DESeq, DESeq2, DEseq2, Design Expert, Design-Expert, DIAMOND, Discovery Studio, DIVA, DNAMAN, DnaSP, Docker, dplyr, DSSP, E-Prime, eBURST, Eclipse, edgeR, EdgeR, EEGLAB, EMBOSS, EndNote, Endnote, Enrichr, ENSEMBL, Ensembl, Ensemble, Epi Info, EpiData, Epidata, ESPript, ESTIMATE, Excel, excel, Excel¬Æ, FACS Diva, FACSDiva, FastQC, FASTQC, FastTree, FCS Express, Feature Extraction, featureCounts, FieldTrip, FigTree, Fiji, FIJI, FIMO, FLASH, Flowjo, FlowJo, Fluidigm, FMRIB, FoldX, FreeBayes, FreeSurfer, Freesurfer, FSL, FSTAT, g:Profiler, G*Power, Galaxy, GATK, Gaussian, Gblocks, GCTA, Geant4, GEMMA, GenAlEx, Gene Set Enrichment Analysis (GSEA), GENECONV, Geneious, Geneious Prime, GeneMANIA, GeneMapper, GeneMark, GenePix Pro, GENEPOP, Genepop, GeneSpring, GeneSpring GX, Genevestigator, Genomatix, GenomeStudio, geNorm, GeNorm, GenStat, GEO2R, Gephi, GEPIA, ggplot2, GIMP, GIS, GISTIC, Glide, Glimmer, glmnet, GOLD, Google Forms, gplots, Grad-CAM, Graph Pad, Graph Pad Prism, GraphPad, Graphpad, GraphPad InStat, GraphPad PRISM, Graphpad Prism, GraphPad Prism, GraphPad prism, GraphPad Prism¬Æ, GREAT, GROMACS, Gromacs, GSEA, GSVA, HADDOCK, Hadoop, HaplotypeCaller, Haploview, HHpred, HISAT2, Hisat2, HKL2000, HMMER, HMMER3, HOMER, HTSeq, HyPhy, I-TASSER, IDEAS, Igor Pro, igraph, IGV, image J, Image J, Image Lab, Image Pro Plus, Image Studio, Image Studio Lite, Image-J, Image-Pro Plus, ImageJ, imageJ, ImageLab, ImageQuant, ImageQuant TL, Imaris, IMARIS, IMOD, IMPUTE2, Inception, Ingenuity Pathway Analysis, Ingenuity Pathway Analysis (IPA, Ingenuity Pathway Analysis (IPA), Inkscape, Integrative Genomics Viewer, Integrative Genomics Viewer (IGV), InterProScan, InterVA, IPA, IQ-TREE, iTOL, JAGS, Jalview, JASP, Java, JBrowse, jModelTest, Jmol, JMP, JMP Pro, JoinMap, Jupyter, Kallisto, Kaluza, Keras, KNIME, KOBAS, LabChart, LabVIEW, LabView, Labview, Lasergene, LAST, LCModel, LEfSe, Leica Application Suite, LIBSVM, LigPrep, LIMMA, limma, Limma, LinRegPCR, LiST, Living Image, lme4, lmerTest, LORETA, MACS, MACS2, MAFFT, MAGIC, MAGMA, MAKER, MapChart, MapMan, MapReduce, MARS, MAS, MASCOT, Mascot, Mask R-CNN, MASS, MassLynx, Mathematica, MatInspector, MATLAB, Matlab, MatLab, MATLAB¬Æ, Matlab¬Æ, Matplotlib, matplotlib, Mauve, MaxEnt, Maxent, MAXQDA, MaxQuant, MCODE, MCScanX, MedCalc, MEGA, Mega, MEGA X, MEGA4, MEGA5, MEGA6, MEGA7, MegAlign, MEGAN, Mercury, Mesquite, MetaboAnalyst, MetaCore, metafor, MetaMap, MetaMorph, Metamorph, Metascape, MeV, Mfold, mfold, MG-RAST, mgcv, Microsoft Access, MIM, Mimics, Minitab, MIRA, miRanda, miRDB, miRDeep2, miRWalk, MISA, ModelFinder, MODELLER, Modeller, Modeltest, ModFit, ModFit LT, MOE, MolProbity, Monocle, Moodle, MOTHUR, Mothur, mothur, Mplus, MrBayes, MS Excel, multcomp, MuMIn, MUMmer, MUSCLE, Muscle, Mutation Taster, MutationTaster, MuTect, mVISTA, MySQL, NAMD, NEST, NetMHCpan, NetNGlyc, Neurolucida, NEURON, Newbler, NIH Image, NIH Image J, NIH ImageJ, NIS Elements, NIS-Elements, nlme, NMDS, NONMEM, NormFinder, Normfinder, NumPy, NVivo, Nvivo, NVIVO, Odyssey, Office Excel, OpenCV, OpenMP, OpenSim, Origin, Origin Pro, OriginPro, ORTEP, ORTEP-3, OrthoFinder, OrthoMCL, OsiriX, PAM, PAML, PANTHER, Panther, Partek Genomics Suite, PartitionFinder, PASS, PAST, PASW, PASW Statistics, Pathway Analysis, Pathway Analysis (IPA, Pathway Studio, PATRIC, PAUP, PAUP*, pClamp, pCLAMP, PDQuest, PennCNV, Perl, Perl script, Perl scripts, Perseus, PHASE, Phaser, PHASER, PHASTER, pheatmap, PHENIX, Phenix, phenix, Phobius, Photoshop, Phylip, PHYLIP, PhyloBayes, phyloseq, PhyML, PHYML, Phyre2, Picard, PICRUSt, PicTar, Pilon, PISA, PITA, PLATON, PLINK, Plink, PolyPhen, Polyphen, PolyPhen-2, PolyPhen2, PostgreSQL, PowerPoint, Presentation, PRIMER, Primer Express, Primer Premier, Primer-BLAST, Primer3, PRISM, Prism, Prism GraphPad, pROC, PROCHECK, Prodigal, Prokka, Prot√©g√©, ProteinPilot, Proteome Discoverer, ProtParam, ProtTest, PROVEAN, PSI-BLAST, PSIPRED, PSORT, PSORTb, psRNATarget, Psychophysics Toolbox, publCIF, PyMol, PyMOL, Pymol, PyMOL Molecular Graphics System, Python, PyTorch, QGIS, QIIME, QIIME2, Qualtrics, Quantity One, QUAST, R, R package, R package vegan, R script, R scripts, R Studio, R studio, R-package, R/Bioconductor, Random, Random Forest, randomForest, RAPID-AUTO, RaptorX, RAST, RaxML, RAxML, RDKit, RDP, Reactome, RED, REDCap, RefFinder, REFMAC, REFMAC5, RELION, RepeatMasker, RepeatModeler, ResFinder, REST, Review Manager, REVIGO, RevMan, rms, RNAfold, RNAhybrid, RNAmmer, RNAz, ROS, Rosetta, RSEM, RStudio, Rstudio, SADABS, SAGE, SAINT, SAINT-Plus, SAMOVA, SAMtools, samtools, Samtools, SAS¬Æ, SaTScan, Scaffold, SCALEPACK, Schr√∂dinger, scikit-learn, Scikit-learn, Scion Image, SciPy, scipy, SDS, SeqMan, Sequencher, SEQUEST, Sequest, Seurat, SHELXL, SHELXL97, SHELXS97, SHELXTL, SIFT, Sigma Plot, SigmaPlot, Sigmaplot, SigmaStat, SignalP, SIMCA, SIMCA-P, SIMPER, SimPlot, Simulink, SKAT, Skyline, Skype, sLORETA, SMART, SMOTE, SNAP, SnpEff, SOAPdenovo, SOLAR, SolidWorks, SOM, SPAdes, Spark, SPARQL, Spike2, SpiNNaker, SplitsTree, SPM, SPM12, SPM8, SPSS, SPSS Statistics, SPSS¬Æ, SPSS¬Æ Statistics, SQL, SQLite, ssGSEA, STAMP, STAR, Stata, STATA, Stata/SE, STATISTICA, Statistica, Statistical Package for Social Science (SPSS), Statistical Package for Social Sciences, Statistical Package for Social Sciences (SPSS, Statistical Package for Social Sciences (SPSS), Statistical Package for the Social Sciences, Statistical Package for the Social Sciences (SPSS, Statistical Package for the Social Sciences (SPSS), stats, StatView, Statview, StepOne, StringTie, Structure, STRUCTURE, SurveyMonkey, survival, survminer, SVM, SWISS-MODEL, Systat, T-Coffee, TargetP, Targetscan, TargetScan, TASSEL, Taverna, tBLASTn, TBLASTN, TBSS, TBtools, Tensorflow, TensorFlow, TIMER, TMHMM, TNT, TopHat, Tophat, TopHat2, Tophat2, Topspin, Tracer, TransDecoder, TreeAnnotator, TreeMix, TreeView, Treeview, Trimmomatic, Trinity, tRNAscan-SE, U-Net, UALCAN, Ubuntu, UCHIME, UCLUST, UCSC genome browser, UCSC Genome Browser, UCSF Chimera, UMAP, Unicycler, Unity, UPARSE, UPGMA, USEARCH, VarScan, VASP, VCFtools, vegan, Vegan, Velvet, Venny, Vina, VMD, Volocity, VOSviewer, VSEARCH, WebGestalt, WebLogo, WEGO, WEKA, Weka, WGCNA, WinBUGS, WinGX, WinMDI, WinNonlin, Word2Vec, X-AREA, X-tile, Xcalibur, XCMS, XDS, XGBoost, XLSTAT, YASARA, YOLO, ZDOCK, ZEN, Zen, Zoom.

https://doi.org/10.5281/zenodo.7409765

Software citation support in Github (2022)

GitHub code repository added support for software citation using CFF files in July 2021.

By September 2022, over 11,000 project had added a citation files and the generated recommended citation information had been viewed over 1.2m times.



Druskat, S., & Spaaks, J. H. (2022). The Citation File Format: Growing up to enable better software citation. Zenodo. https://doi.org/10.5281/ZENODO.7049934

https://doi.org/10.5281/zenodo.7409765

Software citation support in JATS4R (2022)

“The aim of this JATS4R recommendation is to enable support for the various ways people may cite software. The JATS4R recommendation passes no judgement on whether an individual XML document is conforming to these or not; the idea is to support whatever use cases journals are seeing and authors are using.” NISO JATS4R Software Citations v1.0. (2022) https://doi.org/10.3789/niso-rp-40-2021

```
<ref id="bib3">
      <element-citation publication-type="software">
            <person-group person-group-type="author">
                  <string-name content-type="username">hubgit</string-name>
            </person-group>
            <year iso-8601-date="2020">2020</year>
            <source>GitHub</source>
            <part-title>jats-validator-docker</part-title>
            <ext-link ext-link-type="uri"
      xlink:href="https://github.com/JATS4R/jats-validator-docker">https://github.com/JATS4R/jats-validator-docker
      </ext-link>
      </element-citation>
</ref>
```

https://doi.org/10.5281/zenodo.7409765

FAIR Principles for Research Software (2022)

F: Software, and its associated metadata, is easy for both humans and machines to find.
F1. Software is assigned a globally unique and persistent identifier.
F1.1. Components of the software representing levels of granularity are assigned distinct identifiers.
F1.2. Different versions of the software are assigned distinct identifiers.
F2. Software is described with rich metadata.
F3. Metadata clearly and explicitly include the identifier of the software they describe.
F4. Metadata are FAIR, searchable and indexable.

A: Software, and its metadata, is retrievable via standardized protocols.
A1. Software is retrievable by its identifier using a standardized communications protocol.
A1.1. The protocol is open, free, and universally implementable.
A1.2. The protocol allows for an authentication and authorization procedure, where necessary.
A2. Metadata are accessible, even when the software is no longer available.

I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.
I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards.
I2. Software includes qualified references to other objects.

R: Software are both usable (it can be executed) and reusable (it can be understood, modified, built upon, or incorporated into other software).
R1. Software is described with a plurality of accurate and relevant attributes.
R1.1. Software is given a clear and accessible license.
R1.2. Software is associated with detailed provenance.
R2. Software includes qualified references to other software.
R3. Software meets domain-relevant community standards.

Chue Hong, N. P., et al. (2022). FAIR Principles for Research Software (FAIR4RS Principles). *Research Data Alliance.* DOI: 10.15497/RDA00068

The FAIR4RS Principles and the Software Citation Principles are mutually compatible

https://doi.org/10.5281/zenodo.7409765

Poll:
What is the largest remaining challenge relating to software citation?

https://doi.org/10.5281/zenodo.7409765

Conclusions

- We’ve raised the profile of software citation with many stakeholder groups
- Lots of good work done, starting to have effects
- Tracking effects is still a research challenge
    - Can we determine if software citation increases over time?
    - Is this due to our work? (as opposed to more software being used in published work)
- Other challenges remain
    - What are the best identifiers for software?
    - Where and how metadata should be stored?
        - E.g. Citation File Format vs CodeMeta vs …
    - What should be cited (software paper / software repo / software archive) in practice?
- And citation is just a part of a larger FAIR, open, and reproducible goal

https://doi.org/10.5281/zenodo.7409765

Thanks & Questions

We're happy to answer questions now, or later

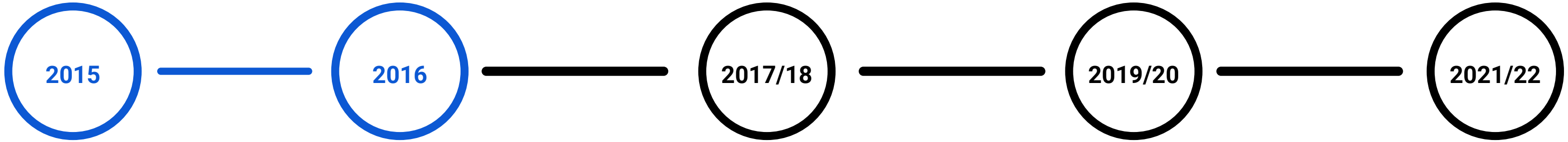
Contact us:

- Neil P. Chue Hong
    - N.ChueHong@software.ac.uk
    - @npch
- Daniel S. Katz
    - d.katz@ieee.org
    - @danielskatz@fosstodon.org

Thank you to all the members of the FORCE11 Software Citation WG and Software Citation Implementation WG.

https://doi.org/10.5281/zenodo.7409765

The FORCE11 Software Citation journey



Software Citation WG started

~55 members (researchers, developers, publishers, repositories, librarians)

Reviewed existing community practices & developed use cases

Software Citation Principles published

Started with data citation principles, updated based on software use cases and related work, working group discussions, community feedback

Software citation principles published after community review: 10.7717/peerj-cs.86

Software Citation Implementation WG started

Group set up to:
1. endorse the principles
2. **develop sets of guidelines for implementing the principles**
3. help implement the principles
4. test specific implementations of the principles

Task forces publish resources

Guidance Task Force develops checklists for authors: 10.5281/zenodo.3479198 and developers: 10.5281/zenodo.3482768

CodeMeta Task Force provides recommendations for schema changes

Journals Task Force starts adoption process with journals and publishers to promote Recognizing the value of software: a software citation guide: 10.12688/f1000research.26932.2

Adoption of software citation increases?

How does software citation fit with other related work:

FAIR for Research Software
Open Research / Open Science
Reproducibility
Software catalogs

"Best Practices for Research Software Registries and Repositories" from Repositories Task Force

Work on backend processing of papers & transfer of citation information to indexers

https://www.force11.org/group/software-citation-implementation-working-group

https://doi.org/10.5281/zenodo.7409765

FORCE11
The Future of Research Communications and e-Scholarship