Science Code Manifesto

A manifesto for science software

Nick Barnes, Climate Code Foundation, 2011-10

Software is a cornerstone of science. Without software, twenty-first century science would be impossible. Without better software, science cannot progress.

But the culture and institutions of science have not yet adjusted to this reality. They need reform to address this challenge. We believe they need to adopt these five principles:

Code

All source code written specifically to process data for a published paper must be available to the reviewers and readers of the paper.

Copyright

The copyright ownership and license of any released source code must be clearly stated.

Citation

Researchers who use or adapt science source code in their research must credit the code's creators in resulting publications.

Credit

Software contributions must be included in systems of scientific assessment, credit, and recognition.

Curation

Source code must remain available, linked to related materials, for the useful lifetime of the publication.

To endorse this manifesto, visit http://sciencecodemanifesto.org/endorse/
To debate it, visit http://climatecode.org/blog/tag/sciencecodemanifesto/

Discussion

Code

All source code written specifically to process data for a published paper must be available to the reviewers and readers of the paper.

The code is the only definitive expression of the data-processing methods used: without the code, readers cannot fully consider, criticize, or improve upon the methods. This is essential to the progress of computational science.

The publishers *must* provide a link, alongside the paper, to a repository containing the code.

The source code made available *should* be the exact version used in processing data for the published paper. Accompanying the source code there *should* be a full description of the platform, language implementation, tools, libraries, and parameters used to run the software.

Reviewing, criticizing, and improving code is easier for readers who can run the code themselves. Use of languages, libraries, systems, and tools which are widely available is strongly recommended.

Copyright

The copyright ownership and license of any released source code must be clearly stated.

Without knowing the ownership and license of the code, readers cannot reuse or derive new works from it, or contribute to its improvement.

This statement *must* be in or alongside the code. If there is no license to adapt the code, that *must* also be stated clearly.

Some source code may be in the public domain - this *must* be stated. Otherwise, code may be owned by authors, institutions, funding bodies, or others. Institutions and funding bodies *should* make clear statements of copyright ownership of research products such as source code: do the copyrights belong to the researchers, the institutions, the funding bodies, some other party, or the public domain?

The terms of any license are up to the copyright owners. An open-source license encourages wide re-use and adaptation, while still allowing conditions such as attribution to be imposed. There are many well-known open-source licenses: use of a well-known existing license is strongly recommended.

Institutions and funding bodies who claim copyright ownership of research products such as source code *should* make clear statements of licensing intent, and *must* make it as simple as possible for researchers to release code under an appropriate license.

Citation

Researchers who use or adapt science source code in their research must credit the code's creators in resulting publications.

They *must* identify the code used, including the specific version, and state its source and ownership. Publishers *must* enforce this through their citation and originality policies. Adapting someone else's code without permission and citation is plagiarism.

The appropriate level of recognition for re-used or adapted code *should* be considered by publishers, and editorial policies *should* be made accordingly. If re-used or adapted code is central to a paper's contribution, co-authorship may be appropriate.

Credit

Software contributions must be included in systems of scientific assessment, credit, and recognition.

Software is an essential research product, and the effort to produce, maintain, adapt, and curate code *must* be recognized. Software stands among other vital scientific contributions besides published papers.

Institutions, funding bodies, professional societies, and other groups *should* review their systems of assessment, credit, and advancement, to give appropriate weight to software contributions. Publishers *should* review their criteria, to encourage publications describing software contributions.

Software development is a complex and valuable skill. Teaching institutions *should* include it in all science degree programs. Research institutions and professional societies *should* include it in their professional development programs. Working researchers *should* consider it an important part of their career progression.

Curation

Source code must remain available, linked to related materials, for as long as a publication remains relevant.

The curator *must* provide the specific version of software used in a publication, along with ownership and licensing information, accessible by a unique stable identifier such as a DOI or URI.

The software *should* be linked to a list of publications using the code, to other versions of the code, to relevant versions of tools and libraries used, and to derived code.

Various providers offer curation facilities, including public open-source repositories and institutional repositories. Institutions *should* make curation recommendations to researchers. Funding agencies *should* require research proposals to include curation plans.

Curators *must* provide a means of reporting and recording software defects and issues, and for communicating those defects to authors and readers. Most public repositories provide a suitable integrated defect tracking system. When a defect is reported, authors *should* assess whether it materially affects their published results. Journal policies on corrections and retractions *should* address the discovery of serious software defects.

Bodies asserting code ownership, and not using open-source licenses, have a particular duty of curation, as they prevent others from voluntarily curating their code.

History

2011-10-08: version 1.0: Changed prefatory text, prepared PDF for website.
2011-07-21: version 0.4: cut out some excess verbiage about defect-tracking, and adjust the curation wording a little.
2011-06-29: tweak layout, add 'history' section.
2011-06-28: version 0.3: add a paragraph about reproducibility and availability of tools, and a line about publication of software contributions, and tweak the introductory wording.
2011-06-15: version 0.2: radical reworking of 0.1: separate into five key areas, each with a terse principle statement. Put these principles up-front, and relegate discussion and description of stakeholder responsibilities to a separate section.
2011-06-14: version 0.1: first draft following Royal Society town hall meeting on open science.