

Complementary Resource List to the Panel Discussion to Improve Visibility of Research Software for Career Advancement

Dr Paula Andrea Martinez and Dr Tom Honeyman, Australian Research Data Commons (ARDC).
DOI: [10.5281/zenodo.6345486](https://doi.org/10.5281/zenodo.6345486)

March 11, 2022

Authors

Dr Paula Andrea Martinez and Dr Tom Honeyman, Australian Research Data Commons (ARDC).
With edits from Jo Savill (ARDC) and Dr Christina Hall (Australian BioCommons).

Recommended citation: Martinez, Paula A, & Honeyman, T (2022). Complementary Resource List to the Panel Discussion to Improve Visibility of Research Software for Career Advancement. Zenodo.
<http://doi.org/10.5281/zenodo.6345486>

DESCRIPTION

This document captures some of the insights shared by panellists and audience participants at a live panel discussion about making research software visible for career advancement. It is a curated summary of questions, commentary, tips and resources that were voiced by panellists and written in the online chat by participants.

EVENT DETAILS

Panel Discussion: Improve visibility of research software for career advancement

Date: Feb 24, 2022

Co-organisers: ABACBS, Australian BioCommons, ARDC Software Program

Watch the recording: <https://youtu.be/OcaLAZ1GdJg>

Research software is not only an essential part of doing research, but it is an important research output. This panel discussion invited four speakers from different career levels, who invest a lot of their time writing, reviewing and maintaining bioinformatics software tools. We had a wide-ranging discussion on how this benefits researchers and others, the effect on career development, research impact via software and what can be done to increase the recognition for this work.

MC: [A/Prof Denis Bauer](#), Principal Research Scientist, Transformational Bioinformatics, CSIRO

Panellists:

- [A/Prof. Kim-Anh Lê Cao](#), NHMRC Career Development Fellow, School of Mathematics and Statistics, The University of Melbourne
- [Dr Sonika Tyagi](#), Central Clinical School, Monash University
- [Mr Fred Jaya](#), School of Life and Environmental Sciences, University of Sydney
- [Professor Gordon Smyth](#), The Walter and Eliza Hall Institute of Medical Research (WEHI)

SUMMARY AND CHAT DISCUSSION

The panel discussion available to watch (<https://youtu.be/OcaLAZ1GdJg>) is complemented by the following summary. Many of the resources mentioned by panellists were summarised in the following questions, commentary, tips and resources shared via the live chat.

Code Review

The issue of code review came up in discussion. Suggestions to ease concerns about errors in code included the importance of thorough testing practices. Code review was also raised as an important feature of creating robust code. One participant asked if it should even be mandatory, while another pointed out that hard policy tends to follow the establishment of normal practice first. The possibility of appointing reviewers specifically for code review was also raised.

As complementary resources, please see: ROpenSci Software Review (<https://ropensci.org/software-review/>) and their open issues (<https://github.com/ropensci/software-review/issues>), the review process of JOSS - The Journal of Open Source Software (<https://joss.theoj.org/>), their review criteria (https://joss.readthedocs.io/en/latest/review_criteria.html) and open reviews (<https://github.com/openjournals/joss-reviews/issues>).

The value of [JOSS](#) was discussed by participants. It was noted that it guarantees code review, is free to submit, submissions are short to write (1-2 pages). It was also pointed out that this doesn't supplant the need to perform other tasks to aid uptake, like documentation, tests and papers, and also that the alignment between paper and software need not be one-to-one.

Software Registries

There were some suggestions and discussion amongst participants about the relative importance of registries. Suggestions and queries specific to bioinformatics included [bio.tools](#), [Dockstore.org](#), [WorkflowHub.eu](#) and [bioconductor.org](#). For other disciplines this [list of domain specific software registries](#) is worth a look at (<https://github.com/NLeSC/awesome-research-software-registries>).

One participant pointed out the importance of attaching identifiers such as DOIs ([Digital Object identifiers](#)) to software. Also, a discussion arose about when to go public with code. One participant proposed the following progression: first use version control locally, then a private repository, then a public one. Questions about establishing precedence arose, and one panellist responded that sharing code after the manuscript is in bioRxiv is what they would opt for, another panellist encouraged to share code via a public repository from day one. A participant noted, to preserve rights as an author, you retain the right to establish terms and conditions when applying a software licence to your shared code.

Software Maintenance

A discussion arose amongst participants about the value and position of maintenance in the overall system, how it is valued and supported, and what can be done to change this. Essential Open Source Software for Science from CZI was mentioned as a specific example supporting some of the Panellists' projects (<https://chanzuckerberg.com/eoss/proposals/?cycle=d-i>), and [Cycle 5](#)

(<https://chanzuckerberg.com/rfa/essential-open-source-software-for-science/>) is open for expressions of interest before April 19, 2022 . The recent Research Software Alliance (ReSA) (<https://www.researchsoft.org/>) report on the funder landscape (<https://doi.org/10.5281/zenodo.6102487>) was also shared.

A panellist first acknowledged that maintenance is an under-supported and under-incentivised activity. The misalignment under the status quo between academic roles and maintenance tasks was mentioned, as was the scale of the task on top of core tasks in an academic role. Participants noted that impact statements can include downloads as a metric in funder reporting. A panellist noted that beyond securing funding, training in maintenance is a significant investment in time, and that maintenance is a major effort for researchers who code.

Software Citation

Software citation, and the mechanisms to make software citable with identifiers was discussed. The Zenodo-GitHub integration (<https://docs.github.com/en/repositories/archiving-a-github-repository/referencing-and-citing-content>) was highlighted as a good method of obtaining a DOI for citation, including DOIs per release. One participant pointed out that citation practice possibly incentivises (re)creating new tools instead of extending existing ones. In response, it was pointed out that new papers can still be written to accompany extensions of existing tools, and that achieving (direct) software citation as a norm will out way that potential mis-incentivisation.

Version Control

Version control systems, their usefulness, and specifically the value of integrations were all discussed. A list of most common version control systems and alternatives was mentioned (<https://alternativeto.net/category/developer-tools/version-control-system/>). The importance of releases on top of versioning (e.g., <https://semver.org/>) was also briefly mentioned.

Commercialisation of software

Commercialisation in regards to open licensing was briefly discussed mentioning one resource <https://opensource.org/faq#commercial>.

SUMMARY OF ATTENDEES

The online panel discussion was attended by 92 people from 167 who registered. The majority of attendees were PhD Candidates, Postdoctoral researchers and Senior scientists or Principal investigators, with 81% of those who registered self-identifying as researchers.