A research institution framework for publishing open code to enable reproducible science

Tom Etherington
etheringtont@landcareresearch.co.nz
@tretherington
Why?

IS THERE A REPRODUCIBILITY CRISIS?

A Nature survey lifts the lid on how researchers view the ‘crisis’ rocking science and what they think will help.

BY MONYA BAKER

52% Yes, a significant crisis
38% Yes, a slight crisis
7% Don’t know
3% No, there is no crisis

1,576 RESEARCHERS SURVEYED
Who will be involved?

- Legal
- Managers
- Scientist programmer
- Research software engineer

Spectrum of speciality and focus

Institution - Science - Code

Intellectual property, risk, preservation of effort, standard operating procedures, reporting
Answering questions, generating knowledge
Coding: guidelines, tools, assistance.
How should code be licensed?

Morin et al. (2012) PLoS Computational Biology 8:e1002598
Where should code be published?

- Have you chosen a permissive or copyleft licence?
- Does your code come with instructions, examples, and tests?
  - Would someone else find your code useful, and be able to use it easily?
- Your code doesn’t link to closed-source or local data or bespoke data formats?
  - Can you install your code on a computer?
  - Are you hoping the code will be developed further?
  - Are you hoping people will let you know about issues and errors?

Mostly yes  Mostly no

Version control hosting services (such as GitHub, GitLab, Bitbucket, GNU Savannah, etc).

and

Archiving services (such as Zenodo, Dryad, FigShare, etc).
How to get credit?

CITATION file or README instructions

(from archiving service)  (Software Sustainability Institute maintains a list)
What standards?

• README file (including: project overview, installation requirements, setup instructions, dependencies),
• LICENSE file (permissive or copyleft),
• citation instructions (either included in the README or as a citation file),
• example data and script,
• documentation embedded within functions,
• good coding practice (such as: commented, indented, white space, logical variable names, function definition),
• sensitive information removed (including: usernames, passwords, application programming interface (API) keys, full paths to files on network drives),
• version control history removed.
What costs?

Money

Time
## Pros and cons

<table>
<thead>
<tr>
<th>Pros:</th>
<th>Cons:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• enables reporting to funders,</td>
<td>• services such as version control hosting and code review procedures need to be administered,</td>
</tr>
<tr>
<td>• better outreach for the scientist and the institution,</td>
<td>• scientists like to have freedom and may prefer to use another platform or like to publish code personally,</td>
</tr>
<tr>
<td>• stimulates collaboration with other researchers,</td>
<td>• ongoing institutional costs associated with version control hosting and archiving services,</td>
</tr>
<tr>
<td>• quality control and transparency of science,</td>
<td>• increased project costs to factor in staff time to make open code well documented and supported,</td>
</tr>
<tr>
<td>• greater visibility of code projects compared to personal accounts,</td>
<td>• accounts and repositories could become abandoned if nobody is using them.</td>
</tr>
<tr>
<td>• provides repository exemplars useful for other scientists,</td>
<td></td>
</tr>
<tr>
<td>• other research institutions are already doing this,</td>
<td></td>
</tr>
<tr>
<td>• provides long-term support or at least access of published code,</td>
<td></td>
</tr>
<tr>
<td>• access and continuity of institutional knowledge if the person</td>
<td></td>
</tr>
<tr>
<td>responsible for a repository leaves the research institution.</td>
<td></td>
</tr>
</tbody>
</table>
A framework for publishing

Roles within research institution:
- Scientist-programmer
- Managers
- Version control service administrators

1. Do science: write code, write paper.
2. Submit paper for internal review.
3. Is open publishing of code appropriate?
   - Yes: Choose a license.
   - No: Don't publish code.

Choose a license:
- Ask yourself these questions:
  - Have you chosen a permissive or copyleft license?
  - Does your code come with instructions, examples, and tests?
  - Would someone else find your code useful, and be able to use it easily?
  - Your code doesn't link to closed-source or local data or bespoke data formats?
  - Can you install your code on a computer?
  - Are you hoping the code will be developed further?
  - Are you hoping people will tell you about issues and errors?

More likely you have software. If you haven't done so already, put your code and supporting documents into a version control service repository, as it will be better published there.

Prepare code for publication.

Approach administrators of your research institution’s version control service account to request a repository, giving them access to view your code.

Does the code meet the standards?
- Yes: Proceed.
- No: Provide feedback and suggest improvements.

Does a version control repository still feel like the logical choice?
- Yes: Open the scientist a repository and give them administrative rights to it.
- No: Move or fork code into the institutional repository.

Archive code in DOI generating archiving service.

Submit paper for publication, citing the archived code DOI and if relevant the repository URL.
A research institution framework for publishing open code to enable reproducible science

Thomas R. Etherington  Ben Jolly  Jan Zörner  Nick Spencer

Abstract

Reproducible science is greatly aided by open publishing of scientific computer code. There are also many institutional benefits for encouraging the publication of scientific code, but there are also institutional considerations around intellectual property and risk. We discuss questions around scientific code publishing from the perspective of a research organisation asking: who will be involved, how should code be licensed, where should code be published, how to get credit, what standards, and what costs? In reviewing advice and evidence relevant to these questions we propose a research institution framework for publishing open scientific code to enable reproducible science.

Keywords: archiving, code, version control, open science, programming
Thank you!

Tom Etherington
etheringtont@landcareresearch.co.nz
@tretherington
But…
Issues identified

1. We don’t really know all possible uses.
2. Outside pressures.
3. We want to encourage people to open up code, and not put bureaucratic obstacles in their way.
4. Do we try to ‘quality control’ and ‘organise’ or just open up completely – will this create a ‘noisy’ or ‘messy’ set of repositories?
5. No one wants to be, or has the time to be, a ‘gatekeeper’.