Changing coding culture

Nature Biotechnology's peer review trial with Code Ocean highlights the importance of 'containers' in enhancing software usability, reproducibility and code-writing in academia.

or the past nine months, *Nature* Biotechnology, together with Nature Methods and Nature Machine *Intelligence*, has piloted a trial with Code Ocean to peer review software accompanying papers. This trial relies on Docker containers—virtual operating systems that bundle code, run scripts, datasets and a computing environment into a single compute capsule, enabling referees to anonymously access and test software without having to worry about configuration and dependency issues. Once a paper is published, the associated capsule is assigned a digital object identifier (DOI) and cited in the article, enabling user access via a link. Although this is not a solution for every software paper, feedback from our trial indicates containers improve the quality, documentation and accessibility of software for both reviewers and users. This bodes well not only for the reproducibility of the science presented in our pages, but also more generally for the quality of code writing in academic research.

The sharing of open-source software is becoming ever more commonplace in biology. All too often, however, the quality of software is less than optimal.

Unlike commercial software—where developers implement formal code review, unit testing and quality assurance of packages—the default for many academic labs is to generate something more akin to 'Soprano code': write it, use it, then 'fuhgeddaboudit'.

Too many academic groups produce programs that are poorly written, organized and annotated. And too few academic groups devote the necessary resources to tackle maintenance and documentation, let alone iron out bugs in their software before submitting their manuscript for review.

As a result, reviewers and readers often face the daunting task of attempting to install and run programs that fall short because no context is provided for the submitted files and code. The problem (see, e.g., *Nat. Biotechnol.* 35, 342, 2017) is how to recreate the authors' original software environment and track down unspecified resources that are necessary for successful code execution—so called dependencies (e.g., the operating system, programming

language, configuration files, run parameters and external code libraries).

This is more than a nuisance for reviewers and users; substandard and/or poorly documented code actively hinders the ability of the research community to establish the reproducibility of a paper's findings.

Nature Biotechnology has peer-reviewed code central to a paper's conclusions for many years. Using services like GitHub, Zenodo or Figshare, reviewers can establish whether code works as advertised (i.e., matches the algorithm described in the paper), evaluate documentation and gauge software accessibility for the broad user community. However, some online environments can pose a threat to referee anonymity, and, as stated above, recapitulating the computing environment of authors' original analyses can present a challenge.

This is where Code Ocean's container platform can help.

A Code Ocean compute capsule recreates the software environment in which an author's original analyses were performed, integrating metadata, code, datasets and software dependencies. It also provides a user-friendly, open-access interface that makes it possible to view or download code, run routines, and save or download output, all with version control. A Code Ocean administrator is available to assist authors in setting up their files in the capsule. Using a custom link, reviewers can anonymously upload their own data and assess the influence of different parameters or code alterations on results.

During the trial, of the 16 papers centered on software selected for external review, 10 teams opted to use Code Ocean. As of 16 April, the journal had published two of these papers linked to capsules (here and here).

Among the authors who declined to participate in the trial, the most common rationale for opting out was that Code Ocean was unsuitable for the code or dataset described in the paper; indeed, many datasets that are >50 GB are currently incompatible with the platform—a potential problem given the increasing size of many biology datasets. The capsule format also

is not suitable when custom code requires supercomputers, specialized hardware or very lengthy running times.

Another reason for deferring from the trial was some authors already had uploaded their code to an alternative site, such as GitHub or BitBucket. While these sites are familiar to computational researchers, nonspecialists are likely to prefer Code Ocean's convenience and user-friendliness.

A third argument for opting out of the trial was the author concern that capsule generation would be a time sink, potentially delaying review and publication. During the pilot, a compute capsule took a median of nine days to be created.

Overall, the feedback from reviewers participating in the trial was positive, and we thank them for devoting their time to this project. But beyond peer review, widespread adoption of Code Ocean—and other container-based services like it, such as Gigantum or Binder—may have a greater role to play in changing the culture in academia around writing code.

For the longest time, much of the effort in academic research has revolved around publications and citations. It is the novelty of an algorithm described in a paper, rather than the accompanying code and optimized software implementation, that has traditionally generated author credit. The fact that Code Ocean provides accreditation for good code (via citation of a DOI) could incentivize better code writing. If researchers start to cite code, journal policies support code citation, and funders and tenure committees recognize code as legitimate research output, then the quality of code writing in academia could truly be transformed.

Submitting software to container services forces people to think systematically about their code, how easily it runs, and what it depends on. Peer review in the container flags issues (such as compatibility and bugs) early on in the development process. This may ultimately reorient academic researchers from writing code to serve themselves to writing code with the user community in mind.

Published online: 3 May 2019 https://doi.org/10.1038/s41587-019-0136-9