

# Easing the burden of code review

*Nature Methods* is one of several Nature journals undertaking a trial with Code Ocean, a cloud-based reproducibility platform, to make it easier to peer review computational code.

An increasing share of modern research relies on analytical code and software. In turn, a good deal of irreproducible research can be attributed to computational tools that are difficult to decipher, use or recreate. Through the concerted efforts of computational researchers and stricter guidelines from publishers, the culture of scientific software is now more open and geared toward dissemination than ever. Yet, even in the era of Git repositories, peer reviewing code can be frustrating and time consuming. Beginning last month, *Nature Methods*, along with *Nature Biotechnology* and *Nature Machine Intelligence*, launched a [trial](#) with [Code Ocean](#) that seeks to facilitate the peer review of computational methods and to improve their reproducibility.

Computational tools are complex objects that depend on many components to run. Dependencies include the operating system, programming language, external code libraries, configuration settings and run parameters. Reproducing these conditions is made even harder by the fact that components typically exist in multiple versions. Many come with their own prerequisites, creating a maddening rabbit hole of dependencies on dependencies.

Since 2014, for papers where custom code is central to the work, Nature Research journals have required that authors include a statement of how their code can be accessed and mention restrictions to access. More recently, these journals have pledged to take a [more active editorial role](#) in encouraging code review.

*Nature Methods* and *Nature Biotechnology* have a long history of [peer reviewing code](#). In our experience, reviews vary widely in their thoroughness, reflecting a number of factors—reviewer workload and resources, the quality of the code and documentation, and whether the reviewer plans to use the code in his or her own research. (Degree of altruism and caffeine may also be factors.) Likewise, numerous problems can surface during the review process. Although this is by no means true for all research software, authors sometimes fail to document all dependencies, and tools can be buggy or can

even require software configurations that are not accessible to reviewers.

Code Ocean aims to improve computational reproducibility by offering a web platform that employs Docker technology to execute code in the cloud. The platform does two key things—it integrates the metadata, code, data and dependencies into a single ‘compute capsule’, ensuring that the code will run—and it does this in a single web interface that displays all inputs and results. Within the platform, it is possible to view or download the code, run routines, and visualize, save or download output, all from a personal computer. Users or reviewers can upload their own data and test the effects of changing parameters or modification of the code.

The platform currently supports more than 11 programming languages commonly used in scientific computing, including Python, R, MATLAB, C/C++, Perl, Julia and Java. It includes GPU support and some preinstalled deep learning frameworks such as Caffe, TensorFlow and Theano. Users can run Jupyter notebook documents that contain executable ‘living’ code and interactive results. Capsules include metadata that are discoverable by web search engines, and critically, Code Ocean provides version control and mints digital object identifiers (DOIs) so that the specific code and environment can be cited directly.

A reasonable worry for authors is the time it takes to generate a compute capsule. On the contrary, it is our hope that this approach will expedite the review of computational papers; one of the goals of our trial is to assess this prediction. Code Ocean aims to ease capsule generation in a number of ways: it includes many preinstalled dependencies, it can import packages from GitHub, and the interface allows drag-and-drop file addition. An administrator guides authors through the process and checks that the code runs on the provided data before making the capsule available to referees.

We recognize that researchers have enormous demands on their time and that reviewing code can add to this burden. We expect that anonymous access to code in

an executable platform will save time and effort by removing the need to install and configure software. As part of this trial, 100 hours of free compute time will be available to each referee per paper.

Reviewing code via a platform like Code Ocean is a step toward easing the burdens of code review and improving reproducibility, but it will not solve every problem. Not every pipeline is compatible with the platform. Code Ocean is not optimized for large datasets (larger than 50–100 GB) at the moment, meaning that preprocessed data may be necessary in place of raw data for large studies. Support for multi-node CPU or GPU clusters is currently not provided.

We stress that the Code Ocean trial is optional and is free for authors and referees, and that referees access the capsules anonymously. Compute capsules that are generated as part of this trial will be linked to the published paper and remain available for users to access. Authors of papers that are reviewed but not ultimately accepted for publication will also have the option to maintain their capsule. Code Ocean has partnered with [CLOCKSS](#) to ensure that capsules are archived in perpetuity.

There are already many examples of well-annotated, well-presented code in computational biology. Nevertheless, expectations in the field are mounting for more researchers to follow best practices. Tools to improve code reproducibility, such as Git repositories, notebook documents, automated workflow annotation tools and container platforms now exist. Our trial will assess whether the Code Ocean platform facilitates peer review for our computational papers. The hope is that it will ultimately help to generate more consistent results for scientists using these methods.

We are very interested in feedback from our readers, authors and referees about their own challenges in computational reproducibility. We encourage you to be in touch with your thoughts on this trial or experiences with it. □

Published online: 31 August 2018  
<https://doi.org/10.1038/s41592-018-0137-5>