

Reviewing computational methods

Assessing papers that report (or use) computational methods is demanding for referees, but peer review of these methods and related software is crucial for biological research.

Two years ago, we released [guidelines](#) for submitting papers describing new algorithms and software to *Nature Methods*. We have continued to publish a good number of such papers since then. In 2014 alone, we published about 50 papers in which an algorithmic development or software tool is central to the work; roughly 98% provide access to software, and at least 75% provide source code.

Easy-to-use software is essential for getting a method into the hands of many scientists. Source code makes the method transparent for developers and allows others to build on the work. Making these available as part of a methods paper is necessary but not sufficient; ideally, both must be explicitly assessed during peer review.

The process of peer review in the biological literature has developed over decades, both at journals and within research communities. But does it work well for biological papers that are computational at heart?

At *Nature Methods* we have been reviewing such papers for some years now. We do not typically encounter resistance when we ask for software and code to be made available. Too frequently, though, our guidelines notwithstanding, papers reporting a computational method are initially submitted without either. Separately, referees quite frequently have trouble running the provided software, which they may report to us during the review process or at the end of it. Both these problems can lead to delays or even to rejection of a paper.

We select referees for their ability to assess code and test software, and we explicitly ask them to do so. Our experience suggests, however, that the level of scrutiny can vary quite widely: some papers receive line-by-line attention to code, whereas in other cases referees assess a piece of software but are brought up short by their failure to run it. Although variable levels of scrutiny are a fact of life in peer review, the process is likely to be particularly challenging for computational papers in the biological literature.

First, the traditional review process is not geared toward troubleshooting software. There is no mechanism for referees and authors to communicate except through the editor, and although this does occasionally happen, it is not an efficient process. In the current publishing model this is hard to change, but releasing software via an accepted repository before submission does allow for troubleshooting and will typically not affect consideration at our journal.

Further, the most useful type of review for a computational methods paper—at least of the sort published at *Nature Methods*—will vary, as these papers fall along a

continuum between a new algorithm and a new software implementation of existing algorithms. On top of this, assessing whether software is usable and works well seems to mean different things to different people—some check for adequate documentation, others go through code, and still others run the software. Without a systematic process in which expectations for referees are made clear, review of such papers is bound to remain variable. We will make improvements along these lines to our review process.

In addition, assessing the general usability of software is difficult. Even if a referee determines that software runs well with the provided sample data, for instance, it might not do so with other data. Factors such as the intuitiveness of the user interface and the ability of a given group to maintain software are quite subjective, yet they are important aspects of whether a method will actually get used.

Finally, there is the question of time. A rigorous review of a computational method with software will typically take longer than that of a more traditional paper. But given the central importance that such methods now hold in so much of biology, evaluating their performance and usability is a critical part of their review.

Authors and journals must do all they can to simplify the process. In addition to providing a way for users to implement and test a method (typically software with code), we ask authors to provide a full description of the method, a user manual and test data, and to specify critical parameters and define the settings used (our full guidelines are [here](#)). It is essential to document the steps involved in generating the data in the paper and to make available the version of the software used. Software should ideally work on all common operating systems. Further, tools now becoming available such as Docker and Galaxy provide a way to test the reproducibility of any computational analysis. Because these tools facilitate the running of software with data sets of choice in a desired environment, they should help with methods assessment, too. So far there is limited use of such tools in review of our papers.

It could be argued that asking referees to review code and run software goes beyond the standards that have been applied in biology until now. After all, we do not ask referees to try to reproduce experimental results in their own labs. But the beauty of computational methods is precisely that this can be done. If we want robust analytical methods that generate believable and reproducible biological results, it should be done.