

Software with impact

The usefulness of computational methods can be improved by releasing code and designing software that supports reproducible research.

Algorithms and the software used to implement them have become an integral and important part of research methods. This is reflected in manuscripts published by *Nature Methods*. Methods with a predominant computational component have increased substantially in number since the early years of the journal, and our expectations for these manuscripts have evolved. In a prior Editorial ("Social software" *Nat. Methods* 4, 189, 2007), we called for greater transparency in custom computer code underlying methods, but also stated that mathematical descriptions or pseudocode are acceptable if an algorithm is sufficiently novel. However, mathematical functions can be converted into software in multiple ways, potentially resulting in different output from the same input. Supplying the full code is strongly preferred because it allows others to see exactly how functions are implemented and it increases reproducibility.

Since 2007 we have published 133 articles with accompanying supplementary software files, of which approximately 70% included the source code. Although many researchers have clearly embraced the idea of releasing their code upon publication—thus increasing the usability, reproducibility and impact of their work—more could do so. *Nature Methods* strongly encourages researchers to take advantage of the opportunity that code repositories, such as GitHub, provide to improve a software tool before submission. Even if others do not examine and test the code, the act of preparing the code and necessary documentation for deposit and use by others will help avoid delays in publication. Too often, editors must postpone peer review while authors prepare the software and documentation; or reviewers are unable to install and run the software owing to undocumented software dependencies. Use of a repository can also improve peer review by streamlining and exposing code revisions.

As with our policy on manuscripts posted to preprint servers such as arXiv, neither deposition nor manuscripts published by other researchers using the software tool will compromise publication of the tool in *Nature Methods*. The tool creators themselves should, however, avoid prior publication of work involving the tool.

But even code release is insufficient when the software is used to acquire new biological results for publication. Nearly all software has user-defined parameters that can or must be tuned to the data characteristics or analysis goals. These parameter values should be reported alongside the output. This practice helps ensure reproducibility even in

the lab doing the work. Software developers can facilitate this process by providing macros to record the software version and parameter settings at the end of an analysis or by integrating such functionality into the software itself. The additional effort this logging requires is minimal compared to the benefits.

Unstable code that fails to provide consistent output can also compromise reproducibility. Here, the application of good software coding practice such as unit testing and automated builds can help ensure that code changes do not have unexpected consequences that alter the output obtained from standard inputs.

When an algorithm or software tool is central to a submitted manuscript, the code must be provided at submission in a sufficiently user-friendly form with full documentation and example test data sets. When relevant, compiled versions should also be provided for testing by potential users. Writing software with a cutting-edge algorithm is very different from providing a tool that works for everyone, and we understand that fulfilling such usability requirements involves substantial time and effort. Use of an established modular open-source software platform to deliver a tool can improve usability while limiting the overall effort required and increasing dissemination.

Often, though, custom software or code is ancillary to a reported method. Furthermore, scientific research, despite being a methodical process, must also be dynamic and suited to exploring risky avenues of investigation. This often necessitates that code be written quickly with no expectation that it will be maintained or used by others in their experiments. But that code is still an integral part of the materials and methods. Just as with biological materials, such code should be made available alongside results obtained using it, with sufficient explanation as to be interpretable. This includes a minimally acceptable level of commenting in the code, a description of software dependencies, and basic installation and use instructions. Because software support is far more burdensome than delivery of materials such as plasmids, this code may be accompanied by a disclaimer that it is released only for reporting and archival purposes.

As with other methods, the impact of computational methods is determined to a great extent by their uptake. An open implementation that supports reproducible research not only provides confidence in the performance of a method but increases the likelihood that other researchers can use and build upon it.