

## Extending transparency to code

Reproducibility initiatives seek to promote greater transparency and sharing of scientific reagents, procedures and data. Less recognized is the need to share data analysis routines. *Nature Neuroscience* is launching a pilot project to evaluate the efficacy of sharing code.

The promise and excitement of the current era of ‘big data’ have often been extolled. To make sense of such datasets, scientists must employ large-scale analyses, often by writing custom software. Concurrently, there is an ongoing effort across scientific disciplines to promote reproducibility, transparency and data accessibility. This has manifested in data-sharing initiatives by governmental and private funders as well as in journals. These efforts to increase reproducibility are laudable and should be supported. Yet, sharing data is of limited value without also sharing the associated analytical software tools. Moreover, sharing of code can increase the impact of an author’s research output and broaden its implementation by others. This applies not only to large collaborative projects but also to those that emerge from individual labs.

Nature Research has encouraged authors to make essential analytical code available upon request since late 2014 (refs. 1,2). As part of that initiative, authors are required to include a statement on how custom code relevant to their publication can be accessed. What we mean by ‘custom code’ is software or algorithms written in the lab and used to generate results. This applies not only to simulations present in computational neuroscience papers but also to analyses in more empirically oriented papers. However, it does not extend to commercial software packages or code used in data collection. In our experience, most authors indicate that the code is available upon request, though there has been an increase in the deposition of code on publicly accessible servers. We support the Commentary published on page 770 of this issue, which advocates sharing code more expansively and outlines best practices for how this should be done<sup>3</sup>. With the publication of this issue, we are also launching a pilot project aimed at evaluating the effectiveness of code-sharing practices.

For the duration of the pilot, authors of papers accepted for publication in our pages will be asked to make the code that supports the generation of key figures in their manuscript available for review. The review will focus on three essential elements for code sharing: accessibility, executability and accuracy. Accessibility can apply to code that is made publicly available (on a server, on the author’s website or as a supplementary part of the manuscript) or distributed privately for the purpose of review. Executability simply means that the code runs without errors (i.e., all functions necessary for running the analyses are included in the deposition). Lastly, the shared and reviewed code should accurately reproduce the key figure(s) in the manuscript.

Authors should make sure to include documentation detailing platform dependencies, the language in which the code is written and the

commands a user needs to run. This also means that authors will have to supply the data on which the code runs. The data need not be in its raw format; it can be preprocessed to an intermediate step as long as all transformations are properly described in the README file. As examples of code sharing, the authors of two recent publications in our pages<sup>4,5</sup> submitted supplementary software files at the time of publication. Both groups agreed to have parts of their code evaluated through our code-sharing procedure. Both passed and we would encourage authors to take a look at these as a guide when developing their own code depositions.

Authors can opt out of this pilot project, although we hope that everyone will participate. At this pilot stage, no paper’s publication will be denied or delayed. It can take some time to ensure that code is well documented, and it is certainly understandable that authors only wish to share code that is error-free. Therefore, we strongly encourage authors to prepare their code-sharing materials early in the manuscript preparation process, for example, while the manuscript is undergoing peer review.

While this project is a preliminary test for *Nature Neuroscience*, it is not without some precedent at Nature Research. Indeed, code review is a part of the full manuscript assessment process at several other journals, including *Nature Methods* and *Nature Biotechnology*<sup>6–10</sup>. We hope that in the future, code and data sharing will become common practice. For example, researchers could adopt open lab notebook practices. Furthermore, journals could aim to host and support infrastructure that enables a reader to access the data that underlie a figure as well as the analytical software used to process the data.

We do appreciate that participating in our pilot to review code sharing will require some further effort from our authors. We don’t take that lightly. But we hope that the steps taken to support code sharing will be viewed as a service to the global research endeavor in terms of promoting reproducibility and transparency. ■

1. *Nature* **514**, 536 (2014).
2. *Nat. Geosci.* **7**, 777 (2014).
3. Eglen, S. *et al. Nat. Neurosci.* **20**, 770–773 (2017).
4. Rosenbaum, R., Smith, M.A., Kohn, A., Rubin, J.E. & Doiron, B. *Nat. Neurosci.* **20**, 107–114 (2017).
5. Barker, G.R.I. *et al. Nat. Neurosci.* **20**, 242–250 (2017).
6. *Nat. Methods* **4**, 189 (2007).
7. *Nat. Methods* **11**, 211 (2014).
8. *Nat. Methods* **12**, 1099 (2015).
9. *Nat. Biotechnol.* **33**, 319 (2015).
10. Evanko, D. Guidelines for algorithms and software in *Nature Methods*. *Methagora* <http://blogs.nature.com/methagora/2014/02/guidelines-for-algorithms-and-software-in-nature-methods.html> (2014).