Cracking the code review process

What does it entail to perform a code review for *Nature Computational Science*?

At *Nature Computational Science*, code is at the heart of all of our papers. For the most part, either the scientific results of a submitted paper are the product of a computational exercise, or a computational method, model, or framework is the main result of the reported research. Thus, perhaps not surprisingly, we see the code as an essential aspect when determining whether a manuscript is suitable for publication with us: for every manuscript that goes to peer review, we do our best to ensure that at least one referee can provide feedback on the corresponding code. Evaluating the code is an important step of the peer-review process, as it reveals not only the extent to which the results are reproducible, but also the level at which the code is complete and reusable. Recently, we shed light on the peer-review process and provided suggestions to aid reviewers in writing constructive and unbiased reviews for *Nature Computational Science*. Here, we would like to suggest some important aspects to consider when peer reviewing the code for a paper.

Before sending a paper to peer review, we ask authors to fill out a code and software submission checklist, which allows them to properly report the details of their code. Our reviewers, in particular those who will be reviewing the code associated with the paper, are encouraged to utilize this checklist, as it provides relevant information, such as where the code can be accessed and/or downloaded. At the moment of peer review, the code might be publicly available, or it might be only accessible to reviewers through our manuscript system or the Code Ocean platform; if and when the paper is accepted for publication, we will work with the authors to deposit their code in a DOI-minting repository and to properly cite it in the paper.

One of the first things to note during the code peer-review process is whether or not the provided code is complete. Ideally, the code and its corresponding input data should be provided for all experiments that are performed in the study, so that they can be properly verified. If this is not the case, we appreciate comments on which parts of the code and data are missing.

As a computational science journal that encompasses many different disciplines, we recognize that different fields may have some limitations when it comes to making code and data available. For instance, in some life and health science studies, the input data can be of a sensitive nature, such as clinical data, meaning that it cannot be shared in its entirety. Ultimately, this may make it difficult to verify if the code runs as expected. In these instances, we will ask the authors to provide a mock dataset so that reviewers are still able to run the corresponding code. In chemistry, materials science, and physics, some studies may use existing software that requires a license, such as Gaussian, Q-Chem, VASP, ABACUS, and COMSOL, among others. In these cases, input files for the licensed software should be shared by the authors so that they can be tested by reviewers. Independent of the situation, whenever parts of the code or the input data cannot be shared, the authors must provide proper justification in the paper and/or in the checklist.

Another important aspect of the code peer-review process is the reproducibility of the results. For each experiment performed in the study, re-running the code provided by the authors should ideally produce results that are consistent with the results reported in the manuscript. If the results obtained when testing the code and data are not similar or consistent with those presented in the paper, we appreciate a detailed description of the test and results in the review.

Finally, we think that it is of utmost importance for the code to be a reusable resource to the community. Practically, this means that detailed instructions should be provided to make the code straightforward to install and to run, not only with the original data, but also with other potential data and parameters. When evaluating reusability during the code peer-review process, it is important to approach the code as a non-expert, meaning that one should avoid making assumptions about instructions or taking steps for granted. It is crucial for the code to be well documented and easy to follow for all members of the community. The referee is also welcome to test the code using their own datasets, in order to evaluate how robust the code is for different data.

While we would appreciate a detailed review including all of these aforementioned aspects of the code, we understand that this can be a very time-consuming task for our reviewers, who already put invaluable effort into reviewing the corresponding manuscript. In addition, reviewers might face limitations when performing the code review, such as lacking the necessary infrastructure or platform for the code execution, or having to wait for a substantial amount of time for the research results to be generated. In these cases, we appreciate any feedback that is possible and feasible: simply checking whether or not the code is complete, and whether or not detailed instructions on how to run and install the code are provided, is already considerably helpful to us.

At *Nature Computational Science*, we strive to make the peer-review process as smooth and informative as possible, and we know that this could not be achieved without all of the outstanding service that our reviewers provide. We hope that these suggestions help make the code peer-review process clearer for our authors and reviewers moving forward. As always, we encourage you to get in touch with us if you have any questions throughout the review process. ❐

Published online: 23 May 2022
https://doi.org/10.1038/s43588-022-00261-w