

But is the code (re)usable?

While it is crucial to guarantee the reproducibility of the results reported in a paper, let us also not forget about the importance of making research artifacts reusable for the scientific community.

The awakening of the scientific community to the reproducibility crisis¹ has stressed the need for more transparency in the reporting of published results. In computational science, among the many different strategies implemented to avoid perpetuating this crisis is the sharing of code and data, with the primary goal of allowing researchers to reproduce the results of a manuscript using their own machines. Reproducibility here means obtaining similar, consistent results using the same code and data as described in the original manuscript². This is certainly an important step that has improved the reporting standards in science, and many journals, including *Nature Computational Science*, ensure that data and code are shared as much as possible and feasible. Nonetheless, researchers often focus solely on reproducibility and may not pay enough attention to another equally important concept: reusability.

Reusability, which can also be referred to as replicability², goes beyond reproducibility: it entails obtaining consistent results with new data, and in some cases, in the context of a new scientific application. Making research artifacts, such as code, reusable allows other researchers to more easily investigate the same or similar scientific questions as new data become available and new ideas are developed, thus helping science progress at a faster pace. For computational research, reusable code may increase the potential impact of the methodology proposed in the corresponding manuscript, as it is more likely that others

will use and/or build on it; non-reusable code, on the other hand, may hinder research progression.

But what is a 'reusable code' in practical terms? A reusable code is accompanied by clear instructions on how to install it, specifying all of the software and hardware requirements. A reusable code does not contain hardcoded settings and hardcoded file names/paths to datasets, which are harder for others to identify and change accordingly; instead, it makes a clear distinction between code and modifiable settings, perhaps by exposing the latter as input parameters of the program. A reusable code includes instructions on how to run it with the original or demonstration datasets, and with other available data, clearly detailing the expected format and output, and any limitations for the execution.

Of course, making sure that code is reusable requires more work than just throwing all of the available code in a public repository in order to comply with sharing policies. That said, the effort of making code reusable certainly pays off when more and more researchers start reusing the code in their own projects, thus increasing its practical impact in the research community. Everyone has tried at least once to reuse an existing code from a paper that either didn't work or that took a substantial amount of time to run because the code was not in a reusable state, and this does nothing but discourage usage and slow down research in general. To put it simply, making code reusable is a noble service to science. If everyone starts a research project

with this mentality, reusability eventually becomes a natural process during software development, and no extra effort is required when sharing code with the community.

We editors at *Nature Computational Science* do our best to ensure that the source codes associated with our papers have the aforementioned elements to make them reusable as much as possible. In addition to asking our referees about reusability during the code peer-review process, we also ask authors to fill out a code and software submission checklist that covers reusability requirements. But we also urge you, the author, to start thinking more about reusability when you are working on a project and submitting a paper. For instance, make sure that you have detailed instructions, making the code easier to install whenever possible, and making it easier to run with new data and parameters as well. A useful tip is to ask a colleague from your lab to try out your code to see whether they can successfully follow the instructions and run your code on a potentially different machine.

Little by little, we can all play a part in creating a better future for science, and soon we will be able to effortlessly stand on the shoulders of giants as we should. □

Published online: 23 July 2021
<https://doi.org/10.1038/s43588-021-00109-9>

References

1. Baker, M. *Nature* 533, 452–454 (2016).
2. *Reproducibility and Replicability in Science* (National Academies Press, 2019).