What is computational reproducibility?

Olivia Guest
Department of Experimental Psychology
University of Oxford, United Kingdom

Nicolas P. Rougier
INRIA Bordeaux Sud-Ouest, Talence, France
Institut des Maladies Neurodégénératives, Université Bordeaux, Centre National de la Recherche Scientifique, UMR 5293, Bordeaux, France
LaBRI, Université de Bordeaux, Institut Polytechnique de Bordeaux, Centre National de la Recherche Scientifique, UMR 5800, Talence, France

Computational modelling is the process by which phenomena found in complex systems are expressed algorithmically. The creation of such simulations is useful because it allows us to test whether our understanding is sophisticated enough to create credible working models of the phenomena we are studying. In neuroscience and cognitive science especially, computational modelling comprises more than just capturing a single phenomenon, it also implements a theory. It gives scientists a method of allowing their ideas to be executed, i.e., for emergent properties to appear when they are implemented and run (McClelland, 2009). In this context, a model is said to be **replicable** if experiments within it can be carried out successfully using the original codebase, with the implicit assumption that such a codebase is available.

However, for models to be evaluated it is mandatory to ensure they are **reproducible** (Topalidou, Leblois, Boraud, & Rougier, 2015). That is, that they can be recreated based on their specification — the details deemed important enough to be included in the accompanying article (Hinsen, 2015). Ideally, this should be possible without contacting the authors for advice, and critically, without referring to the original code (Cooper & Guest, 2014). If the specification is sufficient to successfully recreate the codebase from scratch, then the model is said to be reproducible. This adds further credence to both the model and its overarching theoretical framework. If not, and the model cannot be recreated, then even if the experiments can be carried out successfully within the original codebase, the model is not reproducible (Crook, Davison, & Plesser, 2013).

How to share computational research?

Access to the original codebase is not always straightforward. There have been few substantial changes within scholarly communication and research dissemination since 1665, when the first academic journals (*Le Journal des Sçavans* and *Philosophical Transactions of the Royal Society*) were published. Dissemination of scientific discoveries via publishers continues to consist primarily of static text and figures. However, most research is underpinned by, if not wholly comprised of, code, which is inherently dynamic.

Given code forms the backbone of modern scientific research, it is perhaps unusual that its position within this framework is not clear. For example, it is not straightforward where codebases should be placed: in a footnote (with code assured to be available upon request), in supplementary materials, or in an online repository? Even though more journals are requesting code, as well as raw data, few publisher-backed repositories exist. It is striking that an overwhelming number of journals make no provisions for and offer little guidance on hosting these files or indeed facilitating access to them.

Is it time for progress?

The open source and open science communities proposed solutions to some of the aforementioned problems without publishers' aid nor mediation. Firstly, a set of new innovative software tools (e.g., the binder project) make modelling work more accessible. Secondly, some researchers have taken matters into their own hands and created resources for best practice (e.g., version control: Blischak, Davenport, & Wilson, 2016; Eglen et al., 2016; Wilson, 2016). While others lead by example: Ogrean et al. (2016) published an article with an interactive figure; and the LIGO Open Science Center released extensive amounts of data and code (*LIGO Open Science Center: Tutorials*, 2016). In the same vein, the *ReScience journal* encourages the reproduction of modelling work.

Is the scientific community ready to embrace and facilitate changes with respect to: associating articles with original codebases in a transparent way and, more broadly, making sure computational theories are well-specified and coherently implemented?

References

Blischak, J. D., Davenport, E. R., & Wilson, G. (2016, 01). A quick introduction to version control with git and github. *PLoS Comput Biol*, *12*(1), 1-18. doi: 10.1371/journal.pcbi.1004668

Cooper, R. P., & Guest, O. (2014). Implementations are not specifications: Specification, replication and experimentation in computational cognitive modeling. *Cognitive Systems Research*, *27*, 42 - 49. doi: 10.1016/j.cogsys.2013.05.001

Crook, S. M., Davison, A. P., & Plesser, H. E. (2013). 20 years of computational neuroscience. In M. J. Bower (Ed.), (pp. 73–102). New York, NY: Springer New York. doi: 10.1007/978-1-4614-1424-7_4

Eglen, S., Marwick, B., Halchenko, Y., Hanke, M., Sufi, S., Gleeson, P., ... Poline, J.-B. (2016). Towards standard practices for sharing computer code and programs in neuroscience. *bioRxiv*. doi: 10.1101/045104

Hinsen, K. (2015). Writing software specifications. *Computing in Science & Engineering*, *17*(3), 54–61. doi: 10.1109/MCSE.2015.64

LIGO Open Science Center: Tutorials. (2016). https://losc.ligo.org/tutorials/. (Accessed: 2016-06-02)

McClelland, J. (2009). The place of modeling in cognitive science. *Topics in Cognitive Science*, *1*(1), 11–38. doi: 10.1111/j.1756-8765.2008.01003.x

Ogrean, G. A., van Weeren, R. J., Jones, C., Forman, W., Dawson, W. A., Golovich, N., ... Ebeling, H. (2016). Frontier fields clusters: Deep chandra observations of the complex merger macs j1149.6+2223. *The Astrophysical Journal*, *819*(2), 113. doi: 10.3847/0004-637X/819/2/113

Topalidou, M., Leblois, A., Boraud, T., & Rougier, N. P. (2015). A long journey into reproducible computational neuroscience. *Frontiers in Computational Neuroscience*, *9*(30). doi: 10.3389/fncom.2015.00030

Wilson, G. (2016). Software carpentry: lessons learned [version 2; referees: 3 approved]. *F1000Research*, *3*(62). doi: 10.12688/f1000research.3-62.v2