Verifiable research: The missing link between replicability and reproducibility

Konrad Hinsen[1]

[1]Centre de Biophysique Moléculaire, CNRS, France

April 17, 2023

the WINNOWER

SCIENCE AND SOCIETY

Verifiable research: The missing link between replicability and reproducibility

KONRAD HINSEN [1]

1. Centre de Biophysique Moléculaire, CNRS, France

CORRESPONDENCE:
thewinnower@khinsen.fastmail.net

DATE RECEIVED:
July 08, 2016

DOI:
10.15200/winn.146857.76572

ARCHIVED:
July 15, 2016

KEYWORDS:
#LJAFreproducibility, reproducibility, scholarly communication, computer-aided research

CITATION:
Konrad Hinsen, Verifiable research: The missing link between replicability and reproducibility, *The Winnower* 3:e146857.76572 , 2016 , DOI: 10.15200/winn.146857.76572

To err is human. Scientists being human, they make mistakes. Many if not most of the rules for doing science are designed to weed out mistakes. Reproducibility and replicability are recognized as playing a central role in this process. But a lot of confusion remains about the difference between these two labels and the relation between them. In this essay, I will explain why replicability is the foundation on top of which reproducibility can be constructed, and introduce verifiability as the missing link between them, which deserves particular attention in the context of computer-aided research.

First, a note about terminology. Some people use "reproducible" and "replicable" in the sense I will soon define, whereas others exchange the definitions of the two terms, and yet others seem to consider them synonyms. I hope that the scientific community will ultimately converge to common definitions, but we aren't there yet.

To make the relation between replicability, reproducibility, and verifiability as clear as possible, I will adopt a high-level perspective that intentionally disregards many details, important though they may be in the practice of doing research. For example, when I speak of mistakes, I include in this category everything that may invalidate conclusions drawn from the results of scientific work. This includes forgetting a factor of 2 in evaluating a formula, but also using an insufficient sample size for a statistical analysis, or using an instrument that turns out to be defective. Even fraud is lumped into my general category of mistakes.

A typical report about a scientific study says, "this is what we did, that is what we observed, and those are our conclusions". Replicability is about the observations, whereas reproducibility is about the conclusions. For theoretical and computational work, substitute "this is what we assume, that is what we compute based on our assumptions, and those are our conclusions". Replication is then about the results of the computations. When the computations are done by hand, this is simply called double-checking.

Replication attempts check for mistakes at a technical level. If someone else can independently replicate experimental work or a computation, that means that the published description is sufficiently precise, and it strongly suggests that the original authors did not make a technical mistake in applying their own protocol. A good replication attempt should therefore follow the published instructions as closely as possible. Any deviation from the original protocol makes the interpretation of the outcome more difficult. If the results are significantly different, it is impossible to say if that is due to a mistake in

the original work, a mistake in the replication, or a change introduced in the replication protocol.

Reproduction attempts check for mistakes in the scientific interpretation, and in particular for hidden assumptions. The idea is to perform an experiment or computation that *should*, according to the expectations of experts in the field, lead to the same conclusions, in spite of intentional changes in the protocol. A reproduction attempt therefore retains the important features of the original work but modifies something that according to the current state of knowledge in the field is an unimportant detail. Obviously, reproduction is much more subjective than replication, because what is or isn't important is a matter of personal judgment.

A crucial but often neglected aspect is that *reproduction attempts make little sense unless replicability has been verified first*. The reason is again the possibility to learn something from the outcome. If you try to reproduce a finding and fail, then what? Perhaps you misunderstood the original authors' protocol. Perhaps they made a technical mistake, or you did. Perhaps all the technical work was done correctly but some assumption — yours or theirs — was not justified. If you can replicate their technical work first, you know that a different outcome in a reproduction attempt with a modified protocol is not due to some simple mistake, and thus really adds new scientific insight.

This simple principle is something that I learned as a physics student many years ago, so it hardly counts as revolutionary. In his "Cargo Cult Science" lecture from 1974 [1], Richard Feynman explains it to his students, and cites an anecdote going back to 1937 for an illustration of why it is so often ignored: replication is seen as boring, uninteresting, and unpublishable. Some have even argued that it is useless [2]. With the growing awareness of reliability issues in science, this attitude is finally beginning to change. To cite only two examples, the OSF Reproducibility Project actively supports replication of psychological studies, and the ReScience journal encourages the publication of replication attempts in computational science. There is still a lot more to do — the problems described in [3] are a good example for open technical issues, and we still lack an efficient incentive structure for encouraging replication work — but replicability in science is clearly improving.

Let's imagine an ideal world in which peer review works to perfection, meaning that all published research work has been shown to be replicable at the technical level. Technical mistakes, including fraud, have been effectively eliminated from this world. We can then concentrate on the scientific level, and aim for reproducibility. Ultimately, the conclusions of a study can be considered reproducible if there are many other studies that come to very similar conclusions. It doesn't really matter if those other studies were explicitly designed to be reproduction attempts, or were performed independently and just happen to be similar. We can just look at our nice collection of replicable technical work and its outcomes, and draw our conclusions.

Drawing conclusions implies making scientific judgments, which is always subjective to some extent. But there is also a technical requirement for drawing conclusions, and that is what I call *verifiability*. Even perfectly replicable work is no sound basis for scientific conclusions if it is not verifiable. But we never discuss verifiability explicitly, because until not very long ago, it was simply obvious.

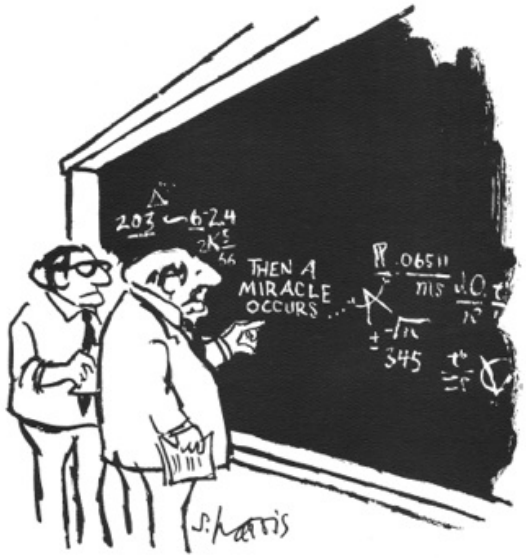

Illustration 1

What non-verifiable work looks like is nicely illustrated in this famous cartoon by Sidney Harris [4]. Even the most superficial reviewer would spot such a manifestly non-verifiable line of reasoning, but subtler cases do end up in the scientific literature. Quite often, reviewers don't take the time to check every argument rigorously if it seems plausible at first sight. But massive widespread non-verifiability, to which reviewers never object, is a recent phenomenon. The ubiquitous modern version of "Then a miracle occurs" is "We used version 2.1 of the program InsightDiscoverer."

The problem with computer software is that even if you can download, install, and run it on your computer, and replicate published results with it, you still do not know what it computes, unless the task and the software are particularly simple. You thus cannot judge if a computation supports a scientific conclusion. In the best imaginable scenario, the software is well documented, so you know what its authors *intended* it to compute. But to err is human. Programmers being human, they make mistakes [5]. With the exception of very simple software that you can completely understand by reading its source code, verifying that a program does what it is supposed to do is nearly impossible. To make it worse, for much of today's complex scientific software there isn't even a complete and precise description — technically called a *specification* — of what it is supposed to do. In the philosophy of science, this problem is called *epistemic opacity* [6]. Unfortunately, the most common attitude in the scientific community today is to shrug off epistemic opacity as inevitable. Journals dedicated to software papers. such as the Journal of Open Research Software or the Journal of Open Source Software, do not even ask reviewers to comment on the correctness of the software's output, because they know that such a request would be unreasonable.

I suspect this is the reason why we use the term "replication", originally applied to experiments, for computations performed by software, instead of the old-fashioned term "double-checking" that we use for manual computations. Double-checking implies verification, because humans cannot do computations well without understanding their context.

Note that this is not just an issue for computational science, i.e. research where computation is the central technique of exploration. Experimental data is processed using software that is often just as opaque as complex simulation software. Worse, we see more and more scientific instruments with integrated computers and embedded software. An increasing part of what we consider raw experimental observations are actually pre-processed by only superficially documented software.

There are numerous examples for this problem, of which I will cite two that I encountered in my own research work in computational biophysics. The theoretical models on which biomolecular simulations

are based are called "force fields". They are complex algorithms that compute a physical quantity called "potential energy" from a graph describing a molecular structure. The published descriptions of these algorithms are not detailed enough to write or verify an executable implementation. Seemingly basic questions such as "Does version 6.2 of the GROMACS software correctly implement the AMBER99 force field?" are therefore meaningless — nobody can say what implementing a force field correctly actually means. My second example is protein structures observed by electron microscopy. The files available from public databases such as EMDB contain numbers whose precise meaning is not explained anywhere. There is the vague notion that higher numbers correspond to a higher local Coulomb potential — the physical quantity that electron microscopes measure – but the exact meaning of the numbers is defined by software that is neither published nor documented.

If we want to maintain the reproducibility of scientific conclusions as a cornerstone of reliable science, we must strive to make computer-aided research verifiable. My own contribution to this is an Open Science project that aims to develop digital scientific notations in which we can express precisely what software is supposed to compute. Everyone is welcome to join, or to develop complementary projects. What matters most is that we stop treating epistemic opacity as a normal and inevitable aspect of using computers.

REFERENCES

1. Feynman RP. Cargo Cult Science [Internet]. 1974. Available from: http://calteches.library.caltech.edu/51/2/CargoCult.htm

2. Drummond C. Replicability is not Reproducibility: Nor is it Good Science. In: ICML 2009 Proceedings [Internet]. Montréal; 2009. Available from: http://cogprints.org/7691/7/ICMLws09.pdf

3. Mesnard O, Barba LA. Reproducible and replicable CFD: It's harder than you think. 2016. Available from: http://arxiv.org/abs/1605.04339

4. Harris S. © ScienceCartoonsPlus.com, reprinted with permission.

5. Soergel DAW. Rampant software errors may undermine scientific results [version 2; referees: 2 approved]. F1000Research 2015, 3:303. Available from: http://f1000research.com/articles/3-303/v2

6. Newman J. Epistemic Opacity, Confirmation Holism and Technical Debt: Computer Simulation in the light of Empirical Software Engineering. In Pisa, Italy; 2015. Available from: http://eprints.bbk.ac.uk/12921/1/12921.pdf