



Publish your computer code: it is good enough

Freely provided working code — whatever its quality — improves programming and enables others to engage with your research, says Nick Barnes.

I am a professional software engineer and I want to share a trade secret with scientists: most professional computer software isn't very good. The code inside your laptop, television, phone or car is often badly documented, inconsistent and poorly tested.

Why does this matter to science? Because to turn raw data into published research papers often requires a little programming, which means that most scientists write software. And you scientists generally think the code you write is poor. It doesn't contain good comments, have sensible variable names or proper indentation. It breaks if you introduce badly formatted data, and you need to edit the output by hand to get the columns to line up. It includes a routine written by a graduate student which you never completely understood, and so on. Sound familiar? Well, those things don't matter.

That the code is a little raw is one of the main reasons scientists give for not sharing it with others. Yet, software in all trades is written to be good enough for the job intended. So if your code is good enough to do the job, then it is good enough to release — and releasing it will help your research and your field. At the Climate Code Foundation, we encourage scientists to publish their software. Our experience shows why this is important and how researchers in all fields can benefit.

Programs written by scientists may be small scripts to draw charts and calculate correlations, trends and significance, larger routines to process and filter data in more complex ways, or telemetry software to control or acquire data from lab or field equipment. Often they are an awkward mix of these different parts, glued together with piecemeal scripts. What they have in common is that, after a paper's publication, they often languish in an obscure folder or are simply deleted. Although the paper may include a brief mathematical description of the processing algorithm, it is rare for science software to be published or even reliably preserved.

Last year's global fuss over the release of climate-science e-mails from the University of East Anglia (UEA) in Norwich, UK, highlighted the issue, and led the official inquiry to call for scientists to publish code. My efforts pre-date the UEA incident and grew from work in 2008 based on software used by NASA to report global temperatures. Released on its website in 2007, the NASA code was messy and proved difficult for critics to run on their own computers. Most did not seem to try very hard, and nonsense was written about fraud and conspiracy. With other volunteers, I rewrote the software to make it easier for non-experts to understand and run. All software has bugs, and we found a number of minor problems, which had no bearing on the results. NASA fixed

them and now intends to replace its original software with ours.

So, openness improved both the code used by the scientists and the ability of the public to engage with their work. This is to be expected. Other scientific methods improve through peer review. The open-source movement has led to rapid improvements within the software industry. But science source code, not exposed to scrutiny, cannot benefit in this way.

NO EXCUSES

If scientists stand to gain, why do you not publish your code? I have already discussed misplaced concern about quality. Here are my responses to some other common excuses.

It is not common practice. As explained above, this must change in climate science and should do so across all fields. Some disciplines, such as bioinformatics, are already changing.

People will pick holes and demand support and bug fixes. Publishing code may see you accused of sloppiness. Not publishing can draw allegations of fraud. Which is worse? Nobody is entitled to demand technical support for freely provided code: if the feedback is unhelpful, ignore it.

The code is valuable intellectual property that belongs to my institution. Really, that little MATLAB routine to calculate a two-part fit is worth money? Frankly, I doubt it. Some code may have long-term commercial potential, but almost all the value lies in your expertise. My industry has a name for code not backed by skilled experts: abandonware. Institutions should support publishing; those who refuse are blocking progress.

It is too much work to polish the code. For scientists, the word publication is totemic, and signifies perfectionism. But your papers need not include meticulous pages of Fortran; the original code can be published as supplementary information, available from an institutional or journal website.

I accept that the necessary and inevitable change I call for cannot be made by scientists alone. Governments, agencies and funding bodies have all called for transparency. To make it happen, they have to be prepared to make the necessary policy changes, and to pay for training, workshops and initiatives. But the most important change must come in the attitude of scientists. If you are still hesitant about releasing your code, then ask yourself this question: does it perform the algorithm you describe in your paper? If it does, your audience will accept it, and maybe feel happier with its own efforts to write programs. If not, well, you should fix that anyway. ■ **SEE NEWS FEATURE P. 775**

Nick Barnes is director of the Climate Code Foundation, Sheffield S17 4DL, UK.
e-mail: nb@climatecode.org

**NOBODY IS ENTITLED
TO DEMAND
TECHNICAL SUPPORT
FOR FREELY
PROVIDED CODE:
IF THE FEEDBACK
IS UNHELPFUL,
IGNORE IT.**

➔ **NATURE.COM**
Discuss this article
online at
go.nature.com/ed3hsl