Challenge to test reproducibility of old computer code

Nicolas P. Rougier, Konrad Hinsen

Putting reproducibility to the test

Konrad Hinsen (Centre for Molecular Biophysics, CNRS, Orléans, France), konrad.hinsen@cnrs.fr
Nicolas Rougier (Inria Bordeaux South West Research Centre, Talence, France), nicolas.rougier@inria.fr
Contact: nicolas.rougier@inria.fr

"Workflows tools can make your computational methods portable, maintainable, reproducible and shareable" as explained in Nature **573**, 149-150 (2019) [1]. But for how long? Will such workflow be still reproducible in a decade? Considering the fast pace of transformations in operating systems and programming languages, it is difficult to foresee future causes of non-reproducibility. This is perfectly illustrated with Python 2 (a very popular programming language in Science) whose official end of life (EOL) has been set to 1st January 2020 in favor of Python 3. Even though this EOL was announced almost a decade ago, this will undoubtedly break things in a foreseeable future.

This naturally questions the long-term validity of computational results if you're unable to run the original code a decade later. This is the reason why we are organizing the ten years reproducibility challenge (https://rescience.github.io/ten-years/) until April 2020 to put reproducibility to the test. We invite researchers to try to run the code they've created for a scientific publication that was published more than ten years ago (before 2010). Sounds easy? We have good reasons to think this might be more difficult than it appears. And maybe the first problem researchers will have to solve is to find their own source code [2]. This challenge should be also an opportunity to identify long-term causes of non-reproducibility.

References

[1] Jeffrey M. Perkel *, Workflow systems turn raw data into scientific knowledge,* Nature 573, 149-150 (2019)
[2] Elizabeth Gibney and Richard Van Noorden *, Scientists losing data at a rapid rate*, Nature News (2013). doi:10.1038/nature.2013.14416