Science-Changing Code

By Benoit Baudry and Martin Monperrus
May 21, 2021

NumPy is a fundamental package for scientific computing with Python [1]. It powers thousands of critical scientific software packages, incl. for analyzing gravitational waves and for pushing the frontiers of cheminformatics [4]. NumPy is on par with sister codes which transformed science [2] and is at the avant-garde of quantum computing [6].

In this post, we give a short overview of NumPy, but from the software engineering side of it [5]: what the code is, how it is developed, how it is executed.

The NumPy source code repository

NumPy started as open source around 2005, as a successor to Numeric, originally created in 1995. It initially built upon code from Space Telescope's Numarray project. The source repository goes back to November 2012, when NumPy started to use the Git version control system. Today, the source code of NumPy is available on the Github code sharing platform [3]. In addition to the source code, this online repository provides the complete history of the development of NumPy, as well as issues, documentation, and statistics about the community.

The number and frequency of changes reflects the depth and the health of NumPy's development. In the last 15 years, the NumPy development team has released 181 versions. This corresponds to one release every month, a remarkable achievement and evidence of a very dynamic development team. Each new release fixes some issues present in the previous one, improves performance or adds new features. The exact contribution of each release is thoroughly documented on the repository, per the best practice of giving good "changelogs" to users. Further evidence of the vibrant development efforts around NumPy is revealed by the number of changes in the code: over time the NumPy development community has contributed 25248 changes, a.k.a commits. Each and every commit has made NumPy more powerful, more reliable and contributed to making NumPy a cornerstone of scientific software.

The public open source repository of NumPy is a remarkable example of a state of the art software development. It can be considered as a role model of engineering foundations for high impact scientific software infrastructure.

The NumPy community

The NumPy community at large is composed of both the developers and the users of the library.

The usage of a version control software enables us to have a rather precise picture of the developer community. According to the Git repository, there have been more than one thousand persons who have contributed at least one change to the NumPy codebase. Per the number of changes, Travis Oliphant is the largest contributor, which corresponds to his recognised role as leader in authoritative sources such as Wikipedia. Yet, it is known that every single contribution matters, and it is the breadth of this collective effort that makes NumPy as reliable and useful as it is today.

It is harder to count the number of users because one can use NumPy without leaving any public trace. The best approximation of the magnitude of the user base is the number of open source programs which explicitly declare that they depend on NumPy. According to Github, there are half a million of such open-source programs.

Testing

NumPy is critical for many applications. Hence, the NumPYyevelopers take a great care of avoiding bugs and regressions. For this, they use software testing, which means they execute the code with certain inputs and verify that the output is correct. Per the best practice in software engineering, those test executions are automated. They are automated along two dimensions as follows. First one can run all tests with a single command, such as "make test". Second, all tests are executed every time there is a change in the NumPy code. In software engineering, the latter process is called "continuous integration": a continuous integration system is a software platform that is responsible for executing all the tests on all changes. The key benefit is to ensure that the change does not introduce a new bug in the program.

Because of the high reliability requirements for NumPy, the NumPy community uses several continuous integration systems. They actually use four of them, which are considered as best-of-breed in software engineering: Github Actions, Travis CI, Circle CI, Microsoft Azure Pipelines. Let us now discuss one of them: Travis CI.

In Travis CI, one execution of all tests is called a "build". The main dashboard of all builds for NumPy is publicly accessible and can be browsed at https://travis-ci.com/github/numpy/numpy/branches. In total since the first activation of Travis CI for NumPy, there have been 35k+ builds. For instance, build #35525 was made for the code change of Bas van Beek made on Jan 22 2021. The execution of all tests took 28 minutes in total. Here is a screenshot of the summary screen for this build.

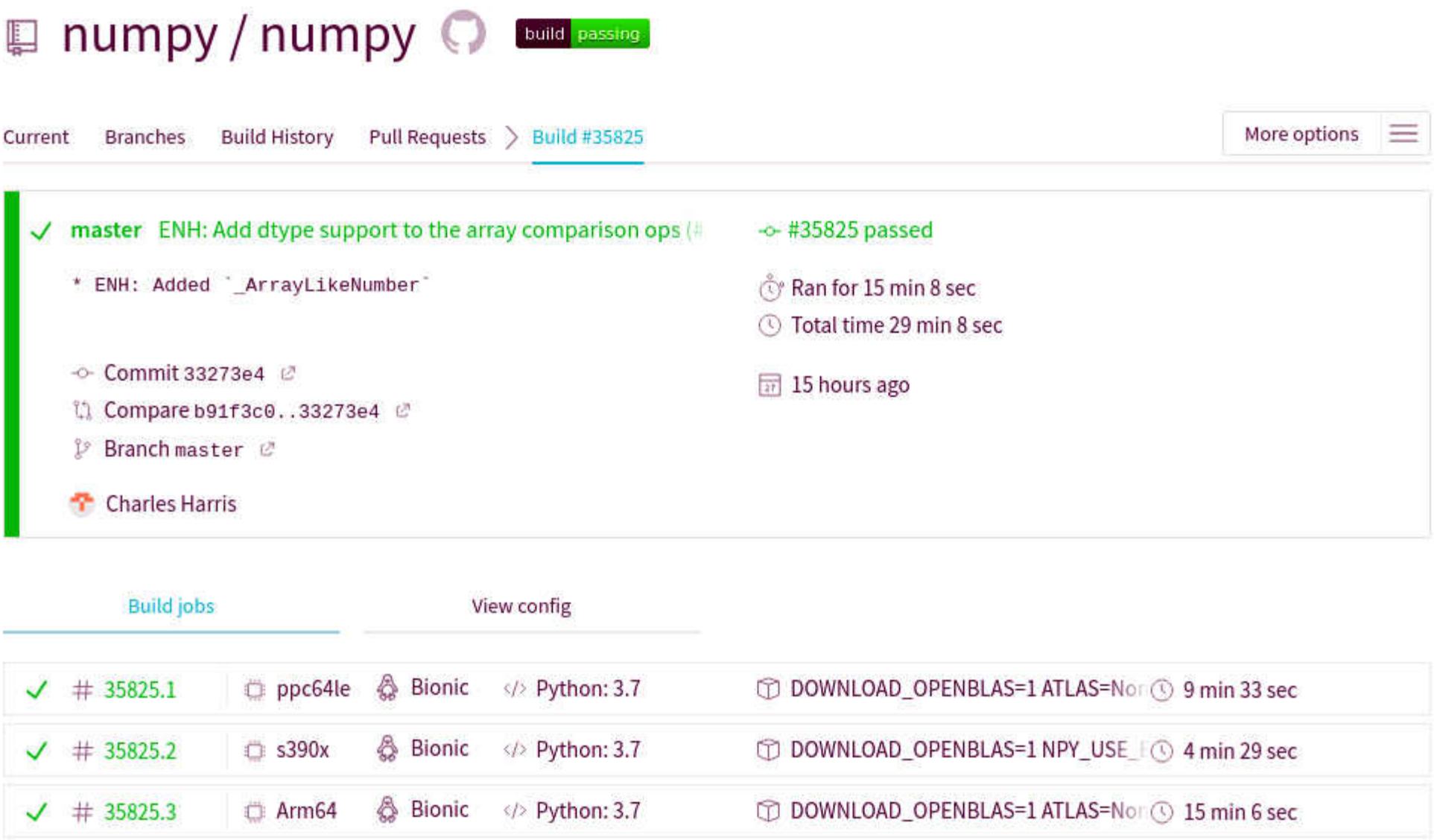

Figure: The recap screen of automated testing for a single change in NumPy.
Credit: travis-ci.org

To recap, NumPy achieves high reliability thanks to automated testing and continuous integration. With such a safety net, it is extremely unlikely that a bug in NumPy would threaten the validity of a major epidemiological simulation or a sound spatialization study.

Conclusion

Computer programs deeply transform science. Beyond every recent scientific discovery there is digital data and programs to analyze, store, make sense of reality through computers. NumPy is one such computer program that has been developed over the last decade, becoming an essential instrument for key discoveries in astronomy, chemistry or biology.

In this post, we gave a gentle introduction on the software engineering methods and tools that the NumPy development team uses. These methods are at the state of the art and at the highest standard of modern software engineering. We believe that following the software engineering best practices has had a deep impact on the reliability and usability of NumPy. With such well-engineered computer programs, it became possible to achieve major scientific breakthroughs, and with no doubt, many high-impact scientific discoveries of tomorrow will be founded on NumPy.

References

[1] The NumPy Website https://numpy.org/

[2] Perkel, Jeffrey M, "Ten computer codes that transformed science," *Nature News Feature*, 2020

[3] The NumPy source code repository https://github.com/numpy/numpy

[4] Harris, Charles R., et al. "Array programming with NumPy." *Nature* 585.7825 (2020): 357-362.

[5] Millman, K.J., Pérez, F. "Developing open source scientific practice," *Implementing Reproducible Research* 149 (2014).

[6] Charles Q. Choi. In the Race to Hundreds of Qubits, Photons May Have "Quantum Advantage," *IEEE Spectrum*, March 2021.

Benoit Baudry *is professor at KTH Royal Institute of Technology, and director of the CASTOR Software Research Center.* ***Martin Monperrus*** *is professor at KTH Royal Institute of Technology, where he holds a Chair from the Wallenberg Autonomous Systems and Software Program.*