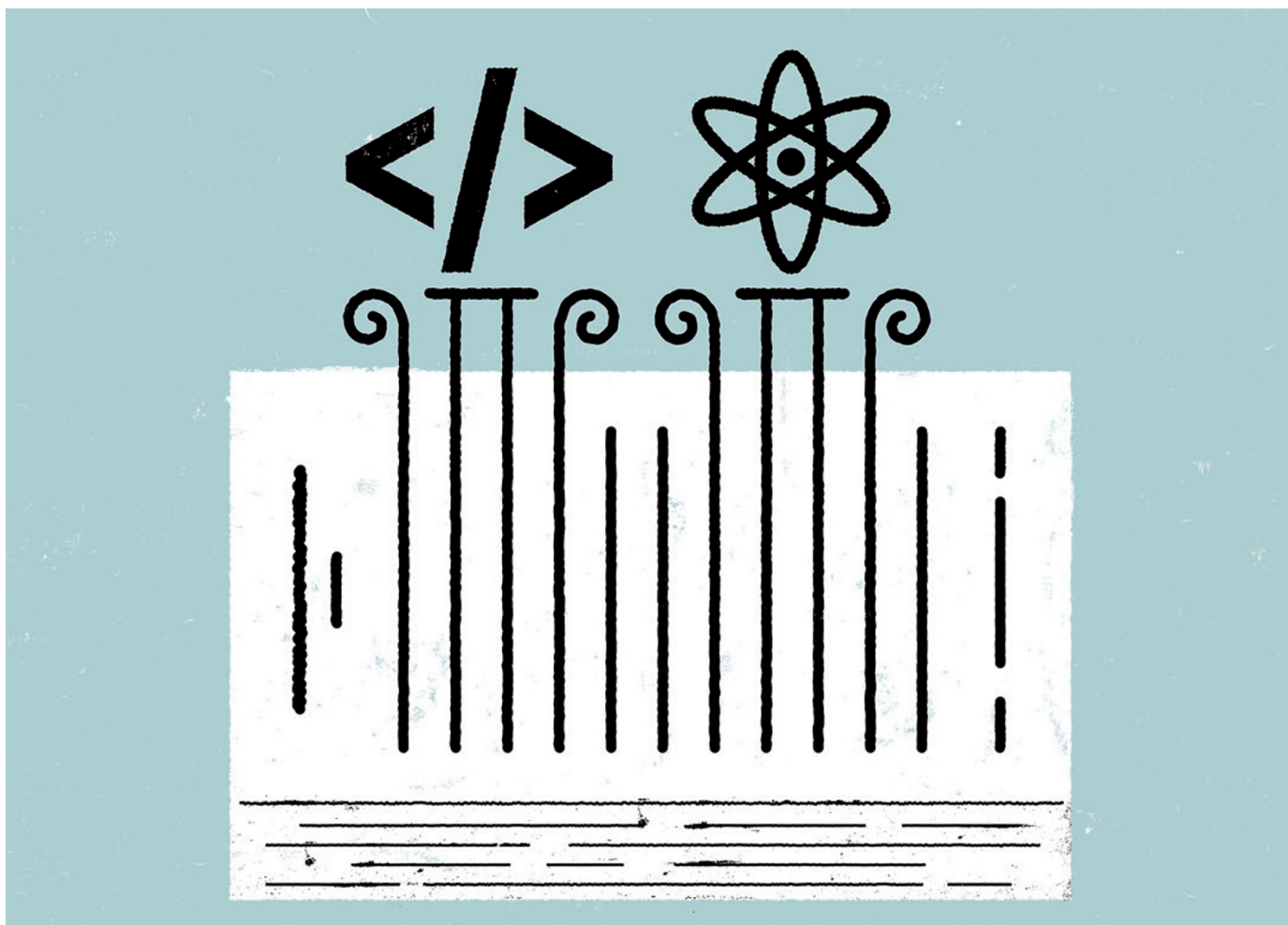


THE UNSUNG HEROES OF SCIENTIFIC SOFTWARE

Creators of computer programs that underpin experiments don't always get their due — so the website Depsy is trying to track the impact of research code.

ILLUSTRATION BY THE PROJECT TWINS



BY DALMEET SINGH CHAWLA

For researchers who code, academic norms for tracking the value of their work seem grossly unfair. They can spend hours contributing to software that underpins research, but if that work does not result in the authorship of a research paper and accompanying citations, there is little way to measure its impact.

Take Klaus Schliep, a postdoctoral researcher

who is studying evolutionary biology at the University of Massachusetts in Boston. His Google Scholar page lists the papers that he has authored — including his top-cited work, an article describing phylogenetics software called phangorn — but it does not take into account contributions that he has made to other people's software. "Compared to writing papers, coding is treated as a second-class activity in science," Schliep says.

Enter Depsy, a free website launched in

November 2015 that aims to "measure the value of software that powers science".

Schliep's profile on that site shows that he has contributed in part to seven software packages, and that he shares 34% of the credit for phangorn. Those packages have together received more than 2,600 downloads, have been cited in 89 open-access research papers and have been heavily recycled for use in other software — putting Schliep in the 99th percentile of all coders on the site by impact. "Depsy ►

► does a good job in finding all my software contributions,” says Schliep.

Depsy’s creators hope that their platform will provide a transparent and meaningful way to track the impact of software built by academics. The technology behind it was developed by Impactstory, a non-profit firm based in Vancouver, Canada, that was founded four years ago to help scientists to track the impact of their online output. That includes not just papers but also blog posts, data sets and software, and measuring impact by diverse metrics such as tweets, views, downloads and code reuse, as well as by conventional citations.

In effect, Depsy recognizes the “unsung heroes” of scientific software, says Jason Priem, co-founder of Impactstory, which is funded by the US National Science Foundation and various philanthropic foundations.

Such a tool is needed, notes Neil Chue Hong, founding director of the Software Sustainability Institute in Edinburgh, UK, because there are few ways to credit scientists for their software. Young researchers are enthusiastic about coding, he says. Last year, he ran a survey of 1,000 randomly selected UK scientists, which suggested that more than 50% of researchers develop their own code. Even so, few UK academics listed code or software as one of their research outputs in the nation’s latest research quality audit (the ‘Research Excellence Framework’) even in disciplines such as computer science that rely heavily on software. “There is a culture that reinforces the idea that producing and publishing code has no perceived benefit to the researcher,” Hong says.

TRACKING SOFTWARE USE

The usual way to track academic impact — by counting citations — still has some relevance to software. Researchers can write papers that describe their software, as Schliep has done for his phangorn package, so that anyone who uses the program can cite it in subsequent papers. But counting citations is an imperfect measure. Researchers may not know which paper to cite, argues Priem, because software packages often have multiple articles associated with them — and some pivotal software projects, he says, such as the GDAL Python library, are not linked to a canonical paper.

If software has no associated paper, there is no universally recognized way to cite it. Still, it is now quite common for coders to assign digital object identifiers (DOIs) to their code, and increasingly to their data sets as well, notes Martin Fenner, technical director of the online repository DataCite in Hanover, Germany. Software is often first stored in the popular code repository GitHub, from which a copy can be automatically archived on scholarly focused repositories such as Zenodo or Figshare, which allocate DOIs to software and thus make it a citable object. Other initiatives are trying to ensure that research papers cite software in a standardized format — such as by

using the Research Resource Identifier.

But counting citations of software DOIs, papers or any other standard format does not reveal the full impact of coders on science, because software so often goes uncited. A 2015 analysis of 90 random biology papers found that two-thirds informally mentioned the use of software, but fewer than half of those papers actually cited the package.

Depsy searches through research papers to discover both citations and informal mentions of software — of which, unsurprisingly, it has found many, says Priem, such as in the acknowledgement sections or the main text of academic papers. But a limitation of the site, Priem admits, is that it currently searches only open-access research papers — missing the vast bulk of paywalled scholarly content. Impactstory will, however, negotiate with publishers for permission to mine the text of paid-access literature.

Mentions in research papers are one of three ways in which Depsy tracks the impact of software, Priem says. Second, the site tracks how code is reused by others. The name Depsy originates from ‘dependency network’ — an overarching term for a map of factors that depend on each other, such as software packages that recycle code from other packages. Depsy calculates the extent to which code is recycled by using Google’s PageRank algorithm, which gives weight to reuse by more-prominent software. From the view of measuring impact, an example of code reuse may be more meaningful than a citation in the literature, Priem notes.

And third, the site gathers download statistics on code packages by trawling through CRAN and PyPI, which are the main repositories for software written in the popular R and Python programming languages, respectively.

FOCUS ON RESEARCH

Other websites do some of what Depsy offers. Crantastic, for example, is a review site that tracks the most popular R packages, and PyPI ranking lists the most popular Python modules by tracking downloads from PyPI. In addition, a few commercial services such as VersionEye and Libraries.io track dependency networks, explaining which software depends on which other packages.

But Depsy is unconventional in its focus on research software, which it distinguishes from other code by identifying key words and the descriptions and titles of software — although the classification process is imperfect, Priem says. The site tracks other code, but it includes research software only when it calculates the percentile impact rankings for academics such as Schliep.

“There is a culture that reinforces the idea that producing and publishing code has no perceived benefit to the researcher.”

Depsy apportions fractional credit to each participant who has contributed to a software package by counting the percentage of code that they have contributed or edited — known in the programming world as a person’s ‘commits’. Fingerprints of each commit are saved in the code, making it easy to track down the originator. But not every edit has the same impact, and Depsy currently cannot distinguish between important contributions and trivial ones. The tool may be adapted to attempt this distinction — by tracking the influence of individual commits — in the future, says Priem.

Depsy also enables users to determine the software with the highest impact in specific disciplines. A search on Depsy for ‘astrophysics’, for instance, yields 11 software packages, of which an analysis and visualization toolkit for astrophysical simulations called ‘yt’ has the greatest impact; it lies in the 97th percentile of all packages.

OBSTACLES TO PROGRESS

One of Depsy’s restrictions, notes Hong, is that it only tracks code that is available in public repositories — so it cannot show the impact of commercial software. Moreover, the site tracks software in only two coding languages: R and Python.

But Depsy’s creators aim to eventually include other coding languages, and to add a fourth way to measure impact: a social-influence metric that would take into account the number of stars that software packages receive from other GitHub users, and how many times a piece of software is discussed online.

The site’s code-reuse metrics have their limitations, too. Researchers often reuse their own code, but might ‘game’ Depsy by repeatedly doing so to garner better profile scores — the software equivalent of citing your own paper. Another way for researchers to game the site might be to start lots of projects but not to finish them, Fenner warns, leaving others to refine them instead; the project originator could then claim credit after the fine-tuned versions of their software become prominent.

“I would love to get to the place where people are trying to game Depsy, because it would mean people are taking software reuse seriously,” Priem says.

Ultimately, transparent metrics that demonstrate the impact of code might enable software creators to secure larger funds during grant reviews, Hong hopes. Science’s coders deserve more funding and support, he says — but getting to that point requires a culture change from everyone involved in scientific research. “The real irony is that by not rewarding the use of software, we’re actually putting roadblocks in the way of science,” Hong says. ■

Dalmeeth Singh Chawla is a science journalist based in London.