Giving software its due

Software and algorithm development is crucial for scientific progress; we discuss how to improve the impact and recognition of these tools.

At *Nature Methods*, we believe that methodological advances drive biology and warrant publications that will reach a broad audience. Among these developments are software tools, which underpin many discoveries, yet often don't get the credit they deserve. Although citations certainly aren't the main impetus for most developers, they help journals and funding agencies get a sense of the need for and uptake of these tools within the community, and can lead to a positive cycle of publications, funding, and further development. Proper citation also improves the reproducibility of experimental results, and thus generally promotes scientific progress.

To learn more about the concerns of the developer community, improve referencing, and share tips for developing tools that have the potential to stand the test of time, we queried developers of popular software tools across different areas of biology for feedback on their experiences and potential best practices. Here we share takeaway points from what we learned.

One of our principal questions was whether developers found that their software tools were usually and correctly cited. The general consensus was that papers describing tools are cited correctly about half the time, but otherwise are either incorrectly cited or not cited at all. A common issue seems to be that high-profile studies that made use of a given tool are often referenced in favor of the paper describing the tool itself. Improper citation might reflect that it's not always obvious how to cite a tool the right way. We recommend that authors state the name of the software and specific release version they used, and also cite the paper describing the tool. In cases when an updated version of a tool used has been published, it should be cited to ensure that the precise version used is clear. And when there is doubt, an e-mail to the developers could resolve the issue.

Proper citation of software developments can be murkier in certain contexts. For example, specific plug-ins and algorithms used within a larger software framework often were developed independently and should also be cited. In addition, algorithms or libraries that are the very basis of successful software and are published independently often get completely overshadowed, at least in terms of citations, by tools that rely on them directly. It becomes easy to see, then, why fair and proper citation can be challenging, even though the vast majority of researchers have the best intentions. Still, it is crucial to address this gap, as proper credit encourages collaboration between people developing tools at different levels to ensure that robust, user-friendly tools ultimately come to light.

Even expert developers who understand the importance of citing such tools said that it is not always easy, and explained that sometimes there is no paper associated with a tool or that they feel they must pick and choose what to cite depending on the reference limits imposed by journals. Still, they provided useful tips for others for getting work cited correctly. Perhaps the most obvious of these was to actually have a reference to cite, be it a preprint or a peer-reviewed paper, and not just a website. They also recommended making it easy for users by pointing explicitly to the relevant information via the readme, the website hosting the software, or even the 'splash screen' that launches the software. Some found it helpful to clearly cite software in presentations that describe or use the tool, and to give software tools a unique name that is easily searchable on the internet.

Best practices in referencing are also an editorial concern, and there are steps we as editors can take to ensure robustness. Apart from looking for problems ourselves, we rely on referees with a deep and up-to-date working knowledge of the relevant tool space to assess whether the literature is adequately documented and the methods are reproducible on the basis of the information provided. However, there is still room for improvement. Several people suggested that journals implement a checklist or reminder for both authors and referees to assess whether the relevant tools are being cited. To this point, we note that Nature Research's Reporting Summary does ask authors to list all software used for data collection and analysis; authors should also cite these tools in the paper. Another fair suggestion was that journals eliminate the cap on the number of citations. We take these points to heart and agree that such steps would improve software citation; we plan to review our policies and workflows accordingly.

In addition to proper referencing, we also asked these developers to share advice for making tools that have the potential to be used widely and have staying power. The major theme among the responses was to develop user-friendly tools that, when possible, are broadly applicable to different user needs. This often requires devoted programmers, who are not always easy to fund. Putting the issue of funding aside, however, they note that the most robust software tools are well documented, are adequately tested, are regularly updated to fix bugs, and reflect the needs of the community they are meant to serve. The last point can require outreach and interaction with biologist users, which can also serve to develop a user base. Making software user-friendly means making it easy to learn, which includes providing tutorials, user guides, and example scenarios.

As is often the case with methods development, a handful of devoted people are often behind tools that benefit thousands. However, unless software tools are in some way linked to a particularly impactful biological research story, the papers describing them might not find homes in prestigious journals. (We consider ourselves an exception, as we celebrate methods for methods' sake.) However, for this reason, metrics like citations are especially important for software developers. Of the scientists queried, almost all said they use numbers of citations as part of grant applications and acknowledge that the popularity of the tools they developed helped them get or keep their jobs.

We value software development as a crucial part of methodological advancement, and we hope that funding agencies that aren't doing so already make it a point to specifically fund programmers and developers. We also thank all of the scientists who took the time to provide us with such excellent feedback. ❐

Published online: 27 February 2019
https://doi.org/10.1038/s41592-019-0350-x