

BIG (DATA) CHANGES

Improving the Evaluation of Biomedical Academic Software Development Projects



With the increasing diversity of assays that produce massive quantities of data, experimental labs are becoming more and more dependent on software tools and online databases, as well as collaborations with bioinformaticians. There is a clear need to attract, train, and support more biomedical data analysts as well as fund more computational “dry-lab” projects.

But throwing more money at software development projects is not enough. The National Institutes of Health (NIH) needs to change how it evaluates such projects. NIH study sections have been optimized for decades to fairly evaluate experimental wet-lab projects. Applying the same approach to computational projects doesn't yield what is really needed: having the most useful software tools and databases continually maintained and enhanced for the long term.

The current NIH review process lacks standards for fair and objective ways to assess the usage, performance, and impact of software tools and databases. Commonly used metrics such as download volume, unique users, and query submissions are currently not required or verifiable. One possible solution is to develop a system similar to Google Analytics where software developers would insert an NIH-certified JavaScript code into their tool and database hosting websites. This code would collect and send user information to a centralized public repository. An alternative would be to develop a global authentication mechanism where users would need to sign in before using NIH supported tools and databases—though this solution may deter users who wish to stay anonymous.

The popularity of tools and databases is not always the best measure of their quality. NIH should seek out more objective benchmarks to assess the quality of algorithms, tools, and databases so the best—those that maximally extract knowledge from the raw data—are selected and recommended.

NIH also needs to find ways to better incentivize the maintenance and enhancement of software tools and databases past their funding term. Currently, useful and popular tools and databases may abruptly disappear upon cessation of funding. Such sudden disappearance can leave wet-bench investigators hanging, without the ability to continue their projects or reproduce their results. Hence, there is a need to develop resources for hosting web-based software applications and databases so that they can remain online and available even after the conclusion of an NIH-supported project.

This can be solved by requiring NIH grant-supported biomedical software developers to provide their tools and databases in self-contained executable environments, such as Docker containers, so that they can be redeployed and hosted in the cloud. In this way, the NIH could cover the low monthly bill of keeping these software services active and available for many years after the funded project has expired. The source code for such projects could also be mandated to be open and placed in codebase repositories for the community to potentially continue to enhance it. Metadata and versioning of tools should also be required for better indexing and provenance.

A fair review of software projects would also consider yet other differences between wet- and dry-lab projects. The life cycle of software projects is shorter than that of typical experimental projects. In addition, to complete software projects, academics often need to hire professional software developers who are presently in high demand and require higher salaries than most NIH-funded researchers can afford. It is also difficult to retain these employees because they are often attracted to work in industry. Big Data science is gradually engulfing biomedical research where computational analysis is becoming the central pillar. Rapid adaptation to these changes is essential, including better management of academic software research development projects. □

The current NIH review process lacks standards for fair and objective ways to assess the usage, performance, and impact of software tools and databases.
