

Establish the missing link between software, data and papers



The Missing Link

Findings presented in papers are based on data and once in a while they come along with data – but not commonly with software. Publication of data has been established already several years ago. Data are published in scientific context either together with a paper or on their own.

This is not standard practice with the related software. But findings are not only based on raw data, they are also based on data obtained in analyses most likely supported by software. So the software used to gain findings plays a crucial role in the scientific work. However, software is rarely seen publishable in terms of scientific publications although it is the link between the findings presented in papers and the data the findings are based on. Thus researchers may not reproduce the findings without the software which is in conflict with the principle of reproducibility in natural sciences.

Although software is made available in various ways, i.e. primarily promoted by solutions originated in the free and open source software movement, the provision lacks solutions serving researchers' needs regarding software used in a scientific context. Making software available like scientific publications would fix the missing link between the interpretation of findings, data and software.

Make software recognised as scientific achievement



Scientific Achievement

Disciplinary journals require that articles discuss scientific questions. Software is often seen only as a contribution to the solution of a question or problem, and not as an independent contribution to science. This means that authors of software must first find a question to motivate the publication in a desired journal.

A direct release of software like a scientific publications is not possible. Thus the scientific achievements of software and its contributions to sciences are poorly perceived and hardly measurable. The resulting gap in interdisciplinary communication regarding scientific software might be closed by software publications in new types of journals, by a common understanding of how to handle scientific software with defined processes, and by commonly accepted and adopted metrics.

Software, which accounts for an increasingly prominent space in research and which has become an indispensable part of science, could be valued and assessed as a contribution to science.

Leverage open access and open science



Open Science

In scientific software development the software and code are not written with the intention for re-use by others. The code is kept and maintained on own computers and servers. If the code grows or groups work together code repositories and version control systems are set up. In many cases these systems then are only available for internal use. These "code repositories" are usually not accessible from the outside. Re-use of code happens primarily on an informal or anonymous base.

Scientists use existing software and code, i.e., from open source software repositories, but only few contribute their code back into the repositories. There are already a number of software platforms available for cooperation and reuse of software, such as SourceForge and GitHub, which are used already by scientists for the provision of scientific software code. These platforms fulfil partly scientific needs to serve software and code in a scientific context as part of the scientific work and scientific tradition. It is unclear, if these platforms can be augmented for scientific purposes or whether special repositories must be created that facilitate not only open source but also open science.

Open code for Open Science means that subsequent users are able to run the code, e.g. by the provision of sufficient documentation, sample data sets, tests and comments which in turn can be proven by adequate and qualified reviews. This assumes that scientist learn to write and release code and software as they learn to write and publish papers.

Establish software engineering rules, best practices and processes in science



Best Practices

For many programming scientists the treatment of source code, e.g. with code design, version control, documentation, and testing is associated with additional work that is not covered in the primary research task. This includes the adoption of processes following the software development life cycle.

The adoption of software engineering rules and best practices has to be recognized and accepted as part of the scientific performance. Most scientists have little incentive to improve code and do not publish code either with their papers or self contained because software engineering habits are rarely practised by researchers or students.

Software engineering skills are not passed on to followers as for paper writing skill. Thus it is often felt that the software or code produced is not publishable. The quality of software and its source code has a decisive influence on the quality of research results obtained and their traceability. So establishing best practices from software engineering not only adopted but also adapted to serve scientific needs is crucial for the success of software publications.