

## A publication model that aligns with the key Open Source Software principles

In recent years, software development has had a significant impact on scientific research and continues to play a major role in facilitating advances with the life sciences in particular. Building code using open repositories such as [GitHub](#) allows it to be continually improved both during the development phase and after the software has been more widely disseminated. However, the long term availability of code is important in order to be reproducible, and to enable future scientific research which may require further modification of existing code<sup>1</sup>. Documentation of code for scholarly purposes usually takes the form of a publication in a peer reviewed article. This allows the developer to provide context around their code for both fellow programmers and non-computational users. A published paper also contributes to the developer's formal academic output but also helps foster vibrant collaborative communities that help nurture and spread new ideas as well as reinforcing the quality of the code that is produced.

Releasing information in incremental steps is nothing new to software developers, who regularly release updates and patches that add new functionality to existing programmes. The launch of a new bioinformatics tool is often accompanied by a paper describing the software for new users. However, the paper describing the tool will be out-of-date as soon as a new software update is released but the changes are often not significant enough to warrant a whole new paper, and thus the most recent developments go undocumented for a sustained period of time. Trying to publish such dynamic information in traditional 'static' journals is much like fitting a square peg in a round hole.

The *F1000Research* (<http://f1000research.com/>) publishing model is much more in synch with the way software is developed. Each software tool published can be updated at any time as a new version (clearly linked to the original and previous versions of the article) allowing any new code, tweaks and features to be documented with relative ease. Furthermore, *F1000Research* ensures that all the code and related data are freely available from the paper. A usable copy of the code as it was at the time of publication remains available, with the code being forked into an archival *F1000Research* space within the same repository used by the authors. A copy of the code as at the time of publication is also assigned a persistent identifier to eliminate any ambiguity about the code that is described in the article. Additionally, *F1000Research* ensures the paper includes a link to the author's own working repository, so that readers can easily navigate to the latest version of the source code. By taking these measures, users are able to establish the provenance of the code and reuse it easily, hence supporting the reproducibility of the software, which ultimately contributes to making the software more robust. *F1000Research* also uses open peer review, providing an additional layer of validation for published software articles. Experts from the scientific community are invited to constructively critique the software and lay the foundations for any improvements. Having these reviews, together with any user comments, open to everyone helps to mirror the collaborative approach encouraged by open source initiatives and embraces the open source 'community' ethos.

By aligning with the requirements of publishing software, *F1000Research* has started to encourage computational science software developers to create an *F1000Research* Article Collection<sup>2</sup> to augment their open source software projects. In February 2014, we launched the BioJS Collection<sup>3</sup> which comprises individual software components, each of which are like a standard Lego-like pieces for building web applications that visualise biological data<sup>4</sup>. With this poster, we will discuss the novel requirements associated specifically with the needs of articles associated with open source software development, and discuss new publishing opportunities that better reflect and support those needs for the benefit of both software developers and scientific researchers as a whole.

1. Prlić A, Procter JB (2012) Ten Simple Rules for the Open Development of Scientific Software. *PLoS Comput Biol* 8(12): e1002802.
2. Markie ML (2014) *F1000Research* Article Collections (<http://blog.f1000research.com/2014/02/13/f1000research-article-collections/>)
3. BioJS Collection (2014) *F1000Research* doi/10.12688/f1000research.collections.2
4. Markie ML (2014) BioJS – visualising biological data: an interview with Manuel Corpas (<http://blog.f1000research.com/2014/02/18/biojs-visualising-biological-data-an-interview-with-manuel-corpas/>)