

Publishing research software

A guide for researchers

Contents

Introduction	2
What is software publishing and when should it be done?.....	2
Ownership and rights	4
Licensing.....	6
Publishing practicalities.....	8
Preserving and sustaining software	9
Publishing a software paper.....	10
Useful links	11
Contacts.....	11
Software publishing checklist.....	12

Many researchers collect experimental data and then use established scientific software, such as Matlab or R, to perform statistical analysis and to visualise the results. It will enable validation of the reported results if you deposit in a data repository both the data themselves and the scripts that executed the analysis performed by the scientific software.

If you write a script to execute a series of operations that process data, for example by converting data from one format to another, it may not be necessary to make the script available, providing the data and metadata are accessible. But it may be of benefit to

this case you should seek the support of the University's IP Management team. They can

owner to reproduce the code will need to be sought, or the code will need to be removed from the published software;

Code downloaded from the Web with no stated licence terms is still someone's intellectual property. If you want to use it, you must seek permission from the provider. If you are unable to secure this permission, you cannot assume you have the right to publish this code.

Where permission to publish software needs to be sought, the Research Data Manager or the IP Management team can advise on the correct way to request this permission.

Licensing

Software, like any intellectual property, should always be distributed under a licence. There are two very good reasons for this:

A licence is an explicit statement of intellectual property rights, and makes it easier to protect these rights;

A licence provides a clear set of permissions and obligations under which the work may be used, so that any users of the work understand what they can and cannot do with it.

Licensing your work need not be difficult or complicated: in most cases the work can be straightforwardly licensed by the application of a standard licence statement. But it does require some forethought to get it right. Two principal considerations should be borne in mind when deciding how to licence your software:

If you do not own the software or any other software with which it integrates, then your options are likely to be already predetermined to a greater or lesser extent. You may be constrained in your licensing options by the requirements of any collaborating parties in the software or the terms of use of any existing software that you have used;

Once you understand any operative constraints on your licensing options, you

the maintenance and development of software by multiple developers, and in addition to code hosting typically provide version control, bug tracking, release management, mailing lists and documentation functionality.

The two approaches can be combined, so that your software is made easily accessible for ongoing use and development via a code repository, while milestone versions of the code, such as those used to generate results reported in an article, are archived for the long-term in a preservation service. For example, GitHub has an integration with Zenodo that enables users to archive a version of a code repository to Zenodo, where the files will be preserved and assigned a Digital Object Identifier (DOI), so that the given version can be easily cited and retrieved.²⁰ If your code is maintained in the University's GitLab or another code repository service, you can archive a version by exporting the code files and documentation and depositing them in your chosen data repository, such as the University's Research Data Archive.

The Software Sustainability Institute provides guidance on choosing a suitable repository for your software and lists a number of resources that can be considered.²¹

Publishing a software paper

Once you have published your software, you may want to use a peer-reviewed article to publish information about it, so that relevant user communities can learn about it and you can receive academic credit for your work. Where the software is a significant output of your research and has potential for re-use, it can be worth publishing a paper that is primarily focused on the software itself, rather than the research questions it was used to solve. There are a number of both general and subject-specific journals which accept submissions that are primarily about software: the Software Sustainability Institute maintains a list of these journals.²²

²⁰ <https://guides.github.com/activities/citable-code/>.

²¹ <http://software.ac.uk/resources/guides/choosing-repository-your-software-project>.

²² <http://www.software.ac.uk/resources/guides/which-journals-should-i-publish-my-software>.

No: make sure your code is clearly and consistently formatted, organised and documented, and is packaged with everything necessary to enable users to read and execute the software.

8. Do I want my code to be used and developed by other researchers or software developers?

Yes: use a suitable software development code repository to publish your software, such as the University GitLab platform or GitHub, and consider publishing an article about your software in a dedicated journal.

No: if the software is required sol