software carpentry

Version Control with Git

Licensing

Learning Objectives

- Explain why adding licensing and citation information to a project repository is important.
- Choose a proper license.
- Explain differences in licensing and social expectations.

When a repository with source code, a manuscript or other creative works becomes public, it should include a file `LICENSE` or `LICENSE.txt` in the base directory of the repository that clearly states under which license the content is being made available. This is because as creative works, source code is automatically eligible for intellectual property (and thus copyright) protection. Code that appears to be, or is expressly advertised as freely available has *not* waived such protection. Hence, those who (re)use code that lacks a license statement do so on their own peril, because the author(s) of the software code can always unilaterally make such reuse illegal.

A license solves this problem by granting rights to others (the licensees) that they would otherwise not have. What rights are being granted under which conditions differs, often only slightly, from one license to another. In contrast to proprietary licenses, the open licences certified by the Open Source Initiative all grant at least the following rights, referred to as the Open Source Definition:

1. The source code is available, and may be used and redistributed without restrictions, including as part of aggregate distributions.
2. Modifications or other derived works are allowed, and can be redistributed as well.
3. The question of who receives these rights is not subject to discrimination, including not by fields of endeavor such as commercial versus academic.

How best to choose an appropriate license can seem daunting, given how many possibilities there are. In practice, a few licenses are by far the most popular, including the following:

- GNU General Public License (GPL),
- MIT license,
- BSD license.

The GPL is different from most other open source licenses in that it is infective: anyone who distributes a modified version of the code, or anything that includes GPL'ed code, must make *their* code freely available as well.

The following article provides an excellent overview of licensing and licensing options from the perspective of scientists who also write code:

> Morin, A., Urban, J., and Sliz, P. "A Quick Guide to Software Licensing for the Scientist-Programmer" PLoS Computational Biology 8(7) (2012): e1002598.

At the end of the day what matters is that there is a clear statement as to what the license is, and that the license is one already vetted and approved by OSI. Also, the license is best chosen from the get-go, even if for a repository that is not public. Pushing off the decision only makes it more complicated later, because each time a new collaborator starts contributing, they, too, hold copyright and will thus need to be asked for approval once a license is chosen.

If the content of a repository contains reseach products other than software, such as data, and/or creative writing (manuals, technical reports, manuscripts), most licenses designed for software are *not* suitable.

- **Data:** In most jurisdictions most types of data (with possibly the exception of photos, medical images, etc) are considered facts of nature, and are hence not eligible for copyright protection. Therefore, using a license, which by definition asserts copyright, to signal social or scholarly expectations for attribution serves only to create a legally murky situation. It is much better to clarify the legal side with a public domain waiver such as Creative Commons Zero (CC0), and the social expectations side with express requests for how to use and cite the data. The Dryad data repository in fact requires this.
- **Creative works:** Manuals, reports, manuscripts and other creative works are eligible for intellectual property protection and are hence automatically protected by copyright, just as software source code. Creative Commons has prepared a set of licenses using combinations of four basic restrictions:
    - Attribution: derived works must give the original author credit for their work.
    - No Derivatives: people may copy the work, but must pass it along unchanged.
    - Share Alike: derivative works must license their work under the same terms as the original.
    - Noncommercial: free use is allowed, but commercial use is not.

Only the Attribution (CC-BY) and Share-Alike (CC-BY-SA) licenses are considered "Open".

Software Carpentry uses CC-BY for its lessons and the MIT License for its code in order to encourage the widest possible re-use. Again, the most important thing is for the `LICENSE` file in the root directory of your project to state clearly what your license is. You may also want to include a file called `CITATION` or `CITATION.txt` that describes how to reference your project; the one for Software Carpentry states:

```
To reference Software Carpentry in publications, please cite both of the following:

Greg Wilson: "Software Carpentry: Lessons Learned". arXiv:1307.5448, July 2013.

@online{wilson-software-carpentry-2013,
  author      = {Greg Wilson},
  title       = {Software Carpentry: Lessons Learned},
  version     = {1},
  date        = {2013-07-20},
  eprinttype  = {arxiv},
  eprint      = {1307.5448}
}
```

Can I Use Open License?

Find out whether you are allowed to apply an open license to your software. Can you do this unilaterally, or do you need permission from someone in your institution? If so, who?