SciLifeLab

DDLS Annual Conference 2025

Workshop 2: Open Source Software for Research – Sharing Code and Software the Right Way

Produced by Alma Nilsson, SciLifeLab Open Science Team
13 November 2025

SciLifeLab

WHAT

Is research code?

is openness for code?

Research Code

Research Code is a ***part of the scientific method*** and should be shared for full transparency.

It can mean many things, what we're talking about today covers all of them.

- **Short scripts** (renaming files, cleaning data, making plots)
- **Pipelines or workflows** (QC or analysis workflows)
- **Interactive notebooks** (for exploring data)
- **Apps and tools** (dashboards, visualisations)
- **APIs or connectors** (linking instruments or transferring data)
- **Software packages or containers** (R packages, Python libraries, Docker images).

Not distinct concepts - the difference often is how it's used

Goble, C. et al. (2020). FAIR Computational Workflows. Data Intelligence, 2(1–2), 108–121. https://doi.org/10.1162/dint_a_00033

FAIR and Open Source

The FAIR principles guide researchers in organising and describing their work so that others, including computers, can find, use, and build upon it.

Open source builds on the same idea: software that is free to access, reuse, modify, and share, without restrictions on who can use it or for what purpose.

- **FINDABLE** – Being able to find a resource , for example through its persistent identifier (e.g. DOI), or through searchable metadata.
- **ACCESSIBLE** – Being able to retrieve (download and open) the resource and its metadata, for example by it being in open file formats and providing clear data descriptions and access conditions.
- **INTEROPERABLE** – Have the resource work for your own system, for example through standardised file formats and vocabularies.
- **REUSABLE** – Understanding it enough to reuse it and legally being allowed to reuse it, through clear licences, documentation, and community standards.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., ... & Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data, 3, 160018. https://doi.org/10.1038/sdata.2016.18
GO FAIR. (2020, February 19). GO FAIR flyer. https://www.go-fair.org/wp-content/uploads/2020/02/GO_FAIR_Flyer_Update19022020-1.pdf
Open Source Initiative. (2024, February 16). The Open Source Definition. Retrieved October 10, 2025, from https://opensource.org/osd

SciLifeLab

WHY do we want open code?

Inaccessible code

1. **Hinders progress** - people can’t build on others work
2. **Can’t verify research results** - lower trust and credibility
3. Reproducing results take **time and resources** (model training)
4. Leads to localised standards and **parallel work**
5. **Personal drawbacks**
    a. Harder to understand your own code later
    b. Wasted time fixing or redoing work
    c. Difficult collaboration and role changes - only one person knows the code
    d. Lost or unclear versions of the software - can't go back to what worked
6. **Harder to get funding and support** (due to openness being widely adopted)

Accessible code

1. **Enables progress through broad participation** - increases access, lowers barriers to innovation, and avoids lock-ins, making science more sustainable.
2. **Ability to verify results** - builds trust within the scientific community and with the public.
3. Saves time and resources
4. **Interoperability** - smoother integration with other tools, data, and workflows.
5. **Software is part of the research method** - without the software, methods are incomplete and shared data can be meaningless.
6. **Personal benefits**
    a. Improved context switching - returning to code is easier
    b. New collaborators can understand
    c. Version control - know what worked and when
7. **Alignment with funder and organisational mandates** - ensures compliance.

The Counter Argument

- **Contents stolen, no recognition –** Use clear licensing and version-tracked repositories. Openness often increases visibility.
- **Too much work** – This also ensures future access and understanding for you. Initial extra work but long-term savings.
- **Too hard** - You don't need to be perfect from the start. Take small steps.
- **My code is too bad to be shared –** No, it's not. Sharing will help you improve.
- **But then I can't sell it** – Open source can be commercialised through support, services, or dual licensing.
- **Open source is low quality** – Not necessarily. Quality depends on practice, not openness.
- **Not everything can be shared safely** - it is guidelines and not absolutes; *as open as possible as closed as necessary.*

SciLifeLab

HOW do we make software FAIR?

Five recommendations for FAIR software

1. Use a publicly accessible [development] repository with version control
2. Add a licence
3. (Register code in a community registry)
4. Enable citation of the software
5. ***Use a software quality checklist***

FAIR-software. (n.d.). Repository recommendations. Retrieved from https://fair-software.nl/recommendations/repository

Tutorial - Sharing Code the Right Way

Now we'll put the concepts into practice.
Using:

1. Your existing (GitHub) repository
2. A new repo for a future project
3. An example repository

We'll apply the **Open Software Checklist** to identify issues and make improvements.

Let's begin!

https://github.com/almanilsson/sharing-code-the-right-way

SciLifeLab

This presentation draws inspiration from the following resources

- **Open-Source and FAIR Research Software for Proteomics**
  Yasset Perez-Riverol, Wout Bittremieux, William S. Noble, Lennart Martens, Aivett Bilbao, Michael R. Lazear, Bjorn Grüning, Daniel S. Katz, Michael J. MacCoss, Chengxin Dai, Jimmy K. Eng, Robbin Bouwmeester, Michael R. Shortreed, Enrique Audain, Timo Sachsenberg, Jeroen Van Goey, Georg Wallmann, Bo Wen, Lukas Käll, and William E. Fondrie. Journal of Proteome Research 2025 24 (5), 2222-2234 DOI: 10.1021/acs.jproteome.4c01079
- **The Turing Way**
  The Turing Way Community. (2022). The Turing Way: A handbook for reproducible, ethical and collaborative research. Zenodo. doi: 10.5281/zenodo.3233853
- **Sharing code**
  Englund, Markus (2023). Sharing code. National Bioinformatics Infrastructure Sweden (NBIS). Educational resource. https://doi.org/10.17044/scilifelab.24161676.v2