

Publish code, software and workflows

NBIS Data Management Team
data-management@scilifelab.se

Presented by Markus Englund
Introduction to Data Management Practices course
21 May 2024

Barnes, N. (2010). Publish your computer code: it is good enough. *Nature*, 467(7317), 753–753. <https://doi.org/10.1038/467753a>

WORLD VIEW

A personal take on events

WWW.SERENATKINS.COM



Publish your computer code: it is good enough

*Freely provided working code — whatever its quality — improves programming and enables others to engage with your research, says **Nick Barnes**.*

I am a professional software engineer and I want to share a trade secret with scientists: most professional computer software isn't very good. The code inside your laptop, television, phone or car is

them and now intends to replace its original software with ours.

So, openness improved both the code used by the scientists and the ability of the public to engage with their work. This is to be expected.

Do you have any unpublished code?



- What code do you have that may be possible to publish?
- Who would benefit from having access to code that you make available?
- What are your excuses for not publishing the code?

Transparency and potential reuse



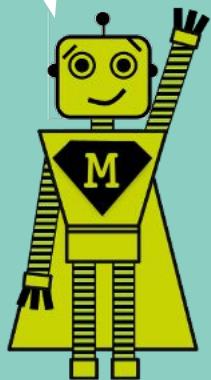
More value to publicly
funded research

Improve peer-review process

Works on my
my computer

High research
integrity

Better research output





“Protocol” & “project plan” icons by Justin Blake, and “infrastructure” icon by Eko Purnomo, from thenounproject.com



Study & data
design

Sampling
& specimen
collection

Sample
preparation

Sample analysis
& data generation

Data processing
to prepare inputs
for analysis

Data
analysis

Communicating
results

Procedures

Biosamples and instruments

Data and computational workflows

Outputs

Information
systems

Laboratory
workspaces

Biobank or
Collection

Data
delivery

Digital
workspaces

Local data
archive etc

Research
databases

What are we talking about?



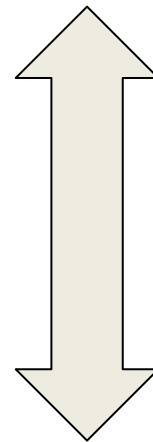
Goble, C. et al. (2020). FAIR Computational Workflows. Data Intelligence, 2(1–2), 108–121. https://doi.org/10.1162/dint_a_00033

code (or source code) – “**set of instructions, or a system of rules, written in a particular programming language**” (Wikipedia)

software – “**set of computer programs** and associated documentation and data. This is in contrast to hardware, from which the system is built and which actually performs the work” (Wikipedia)

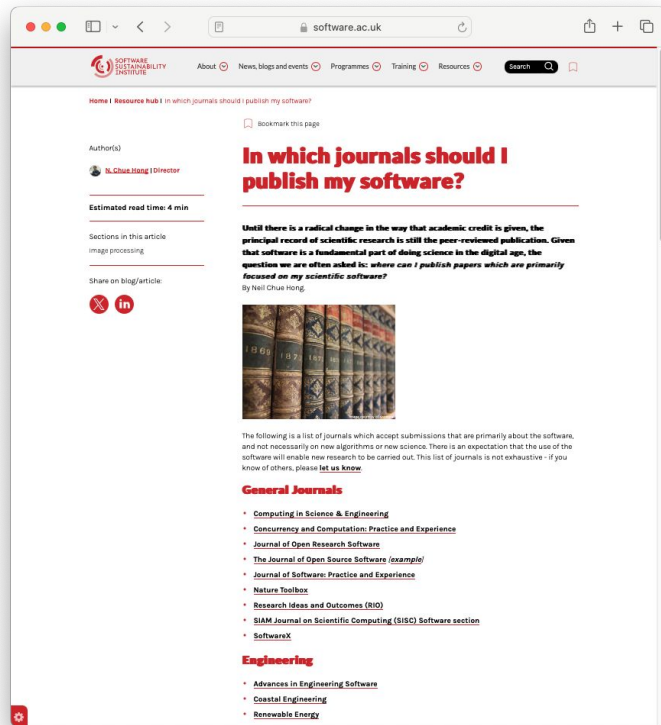
scientific computational workflow – “a multi-step process to coordinate multiple tasks and their data dependencies” (Goble, C. et al., 2020)

**Transparency and
reproducibility**



**Reuse and
shared resources**

In which journals should I publish my software? By the Software Sustainability Institute.



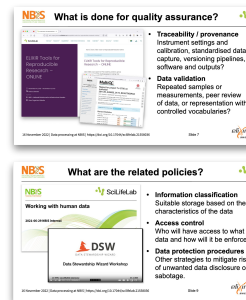
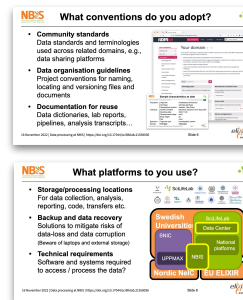
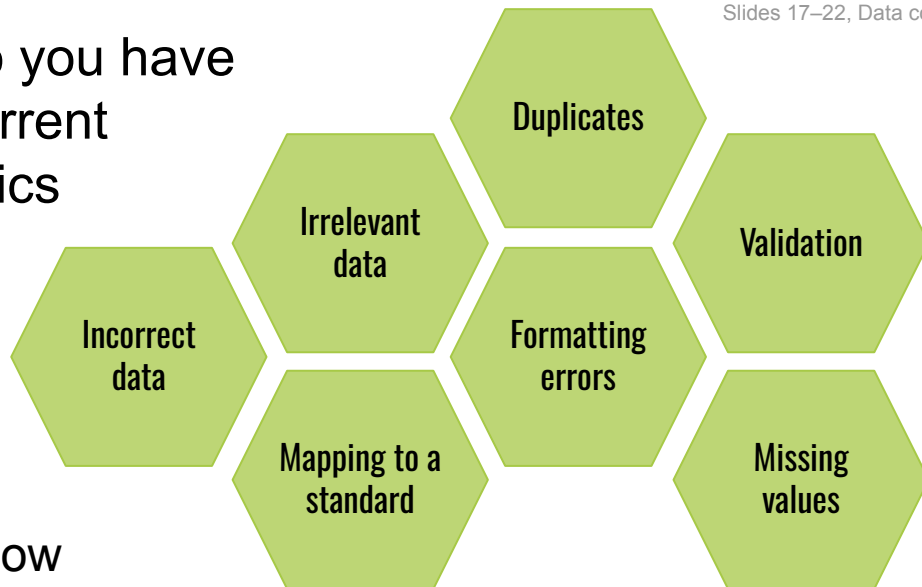
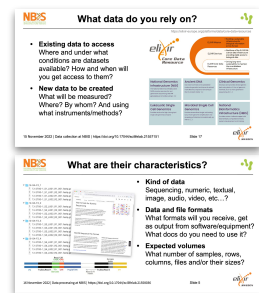
Reproducible analysis



Slides 17–22, Data collection at NBIS, <https://doi.org/10.17044/scilifelab.21557151>

The data do you have
and their current
characteristics

The characteristics that
you want, ready for the
platforms you will use



Typical workflow

Inspecting

Harmonising

Verifying

Reporting/Documenting

Detect unexpected, incorrect,
and inconsistent data, etc.

Fix or remove anomalies,
transform, convert etc.

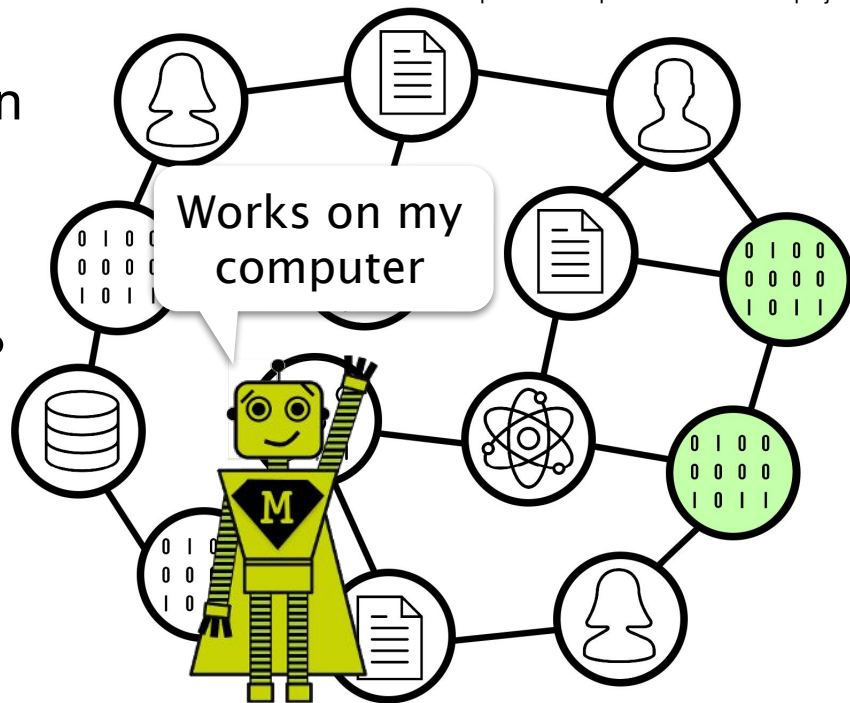
Test that the results are
complete and correct.

Record changes made and
quality assessments.

The FAIR (Findable, Accessible, Interoperable, Reusable) principles can apply to all research outputs

- Where will people be looking?
- Who should have access and how?
- How to package it to make it work for someone else?
- What cross-references and what documentation can you provide?

Graph: "PID Graph" from the FREYA project



Robot: MetaManMachine by Nikola Vasiljevic (2021),
CC BY-SA 4.0, doi:10.5281/zenodo.4471098



Priem, J. (2013). Beyond the paper. Comment in Nature, 495(7442), 437–440. <https://doi.org/10.1038/495437a>

- **Transparency** “This is how we did it!”
- **Reproducibility** – “Run it yourself and verify that you get the same results!”
- **Reusability** – “Use it in new ways!”
- For **describing research methods** in more detail – “See the software’s manual for a more detailed explanation of ...”
- To get more **citations**
- To get **feedback**
- Other?

“We now have a unique opportunity as scholars to guide the evolution of our tools in directions that honour our values and benefit our communities. Here's what to do. First, try new things: publish new kinds of products, share them in new places and brag about them using new metrics.”

– Jason Priem, 2013

Barker et al (2022). Introducing the FAIR Principles for research software. Scientific Data, 9(1), 622. <https://doi.org/10.1038/s41597-022-01710-x>

Article | [Open Access](#) | [Published: 14 October 2022](#)

Introducing the FAIR Principles for research software

[Michelle Barker](#) , [Neil P. Chue Hong](#), [Daniel S. Katz](#), [Anna-Lena Lamprecht](#), [Carlos Martinez-Ortiz](#), [Fotis Psomopoulos](#), [Jennifer Harrow](#), [Leyla Jael Castro](#), [Morane Gruenpeter](#), [Paula Andrea Martinez](#) & [Tom Honeyman](#)

[Scientific Data](#) **9**, Article number: 622 (2022) | [Cite this article](#)

9418 Accesses | **2** Citations | **243** Altmetric | [Metrics](#)

Abstract

Research software is a fundamental and vital part of research, yet significant challenges to discoverability, productivity, quality, reproducibility, and sustainability exist. Improving the practice of scholarship is a common goal of the open science, open source, and FAIR (Findable, Accessible, Interoperable and Reusable) communities and research software is now being understood as a type of digital object to which FAIR should be applied. This emergence reflects a maturation of the research community to better understand the crucial

Download PDF



Sections

References

[Abstract](#)

[Introduction](#)

[Results](#)

[Discussion](#)

[Methods](#)

[Data availability](#)

[Code availability](#)

[References](#)

[Acknowledgements](#)

[Author information](#)



<https://fair-software.nl>

FIVE RECOMMENDATIONS FOR FAIR SOFTWARE

ENDORSE

LET'S GO! →



<https://data.scilifelab.se/services/>

<https://github.com/nf-core/sarek>

Search

Type

- ☐ Community
- ☐ Compute resources
- ☐ Database
- ☐ Helpdesk
- ☐ Portal
- ☐ Storage resources
- ☒ Tool

Services for researchers

Chanjo

Chanjo is a sequencing coverage assessment tool useful in clinical and other sequencing contexts.

Type: Tool

Maintained by: Clinical Genomics Stockholm

Support:

GENMOD

GENMOD is a simple to use command line tool for annotating and analysing genomic variations in the VCF file format.

Type: Tool

Maintained by: Clinical Genomics Stockholm

Support:

Services for data-producing facilities

CheckQC

CheckQC is a program designed to check a set of quality criteria against an Illumina runfolder.

Type: Tool

Maintained by: SNP & SEQ Technology Platform, NGI

Support:

Cutadapt

Cutadapt finds and removes adapter sequences, primers, poly-A tails and other types of unwanted sequence from your high-throughput sequencing reads.

Type: Tool

Maintained by: NBIS

Support:

HPA

Human Protein Atlas

The Human Protein Atlas aims to map all the human proteins in cells, tissues, and organs using an integration of various omics and imaging technologies

Type: Database, Portal, Tool

Maintained by: The Human Protein Atlas

Support:

Tool used as example:

nf-core/sarek

nf-core/Sarek

nf-core/sarek is a workflow designed to detect variants on whole genome or targeted sequencing data.

Type: Tool

Maintained by: nf-core community

Support:

Repository with version control

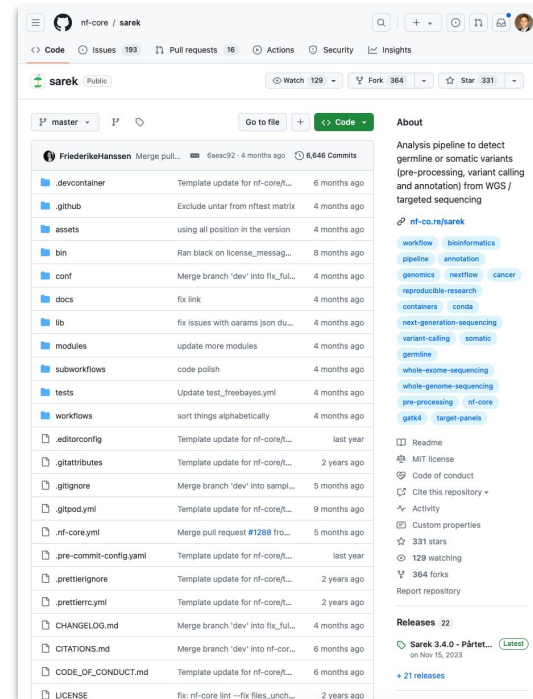
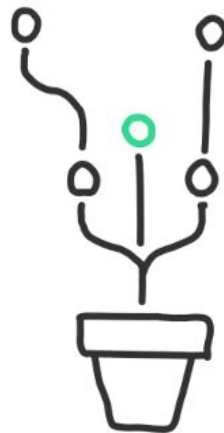


<https://fair-software.nl/recommendations/repository>

<https://github.com/nf-core/sarek>

#1 USE A PUBLICLY ACCESSIBLE REPOSITORY WITH VERSION CONTROL

Such as <https://github.com/>
Many examples of usage at SciLifeLab:
<https://data.scilifelab.se/services/>



<https://fair-software.nl/recommendations/license>

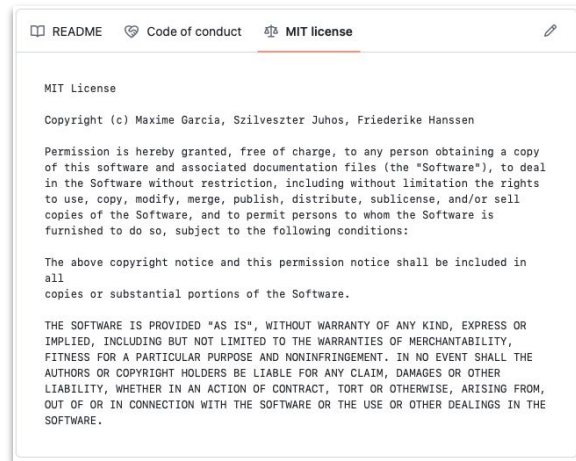
<https://github.com/nf-core/sarek>

#2 ADD A LICENSE

Guidance:

<https://choosealicense.com>

<https://tldrlegal.com>



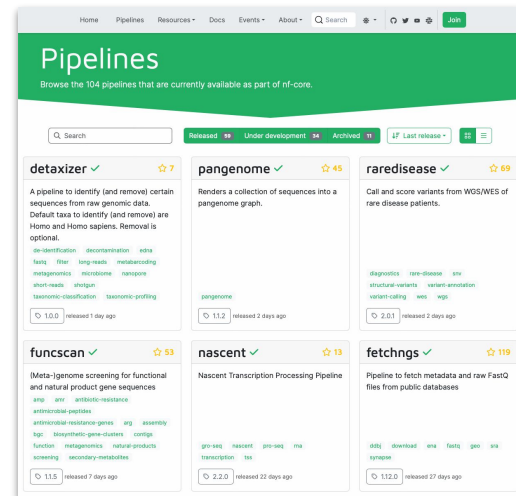
Add a license to a GitHub repository

<https://help.github.com/en/github/building-a-strong-community/adding-a-license-to-a-repository>

<https://fair-software.nl/recommendations/license>

<https://nf-co.re>

#3 REGISTER YOUR CODE IN A COMMUNITY REGISTRY



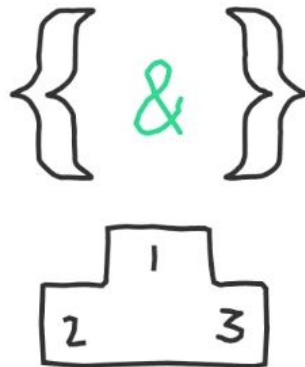
<https://data.scilifelab.se/services/>

<https://nf-co.re/pipelines>

<https://www.bioconductor.org/about/>

<https://github.com/NLeSC/awesome-research-software-registries>

#4 ENABLE CITATION OF THE SOFTWARE



[README](#)
[Code of conduct](#)
[MIT license](#)

Citations

If you use `nf-core/sarek` for your analysis, please cite the Sarek article as follows:

Friederike Hanssen, Maxime U Garcia, Lasse Folkersen, Anders Sune Pedersen, Francesco Lescai, Susanne Jodoin, Edmund Miller, Oskar Wacker, Nicholas Smith, nf-core community, Gisela Gabernet, Sven Nahnsen Scalable and efficient DNA sequencing analysis on different compute infrastructures aiding variant discovery *bioRxiv* doi: [10.1101/2023.07.19.549462](https://doi.org/10.1101/2023.07.19.549462).

Garcia M, Juhos S, Larsson M et al. Sarek: A portable workflow for whole-genome sequencing analysis of germline and somatic variants [version 2; peer review: 2 approved] *F1000Research* 2020, 9:63 doi: [10.12688/f1000research.16665.2](https://doi.org/10.12688/f1000research.16665.2).

You can cite the sarek zenodo record for a specific version using the following doi: [10.5281/zenodo.3476425](https://doi.org/10.5281/zenodo.3476425)

An extensive list of references for the tools used by the pipeline can be found in the [CITATIONS.md](#) file.

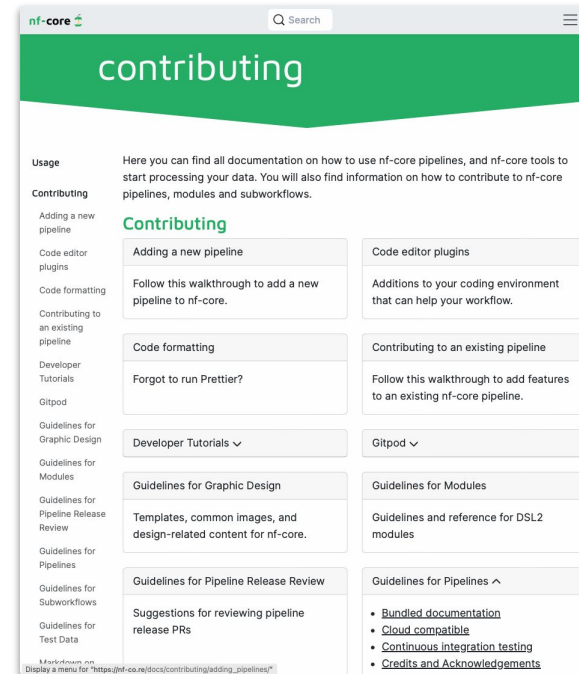
You can cite the `nf-core` publication as follows:

The nf-core framework for community-curated bioinformatics pipelines.

Philip Ewels, Alexander Peltzer, Sven Fillinger, Harshil Patel, Johannes Alneberg, Andreas Wilm, Maxime Ulysse Garcia, Paolo Di Tommaso & Sven Nahnsen.

Nat Biotechnol. 2020 Feb 13. doi: [10.1038/s41587-020-0439-x](https://doi.org/10.1038/s41587-020-0439-x).

#5 USE A SOFTWARE QUALITY CHECKLIST





Curating research artifacts to support scientific integrity.

The CUrating for REproducibility (CuRe) Consortium supports curation of research data and review of code and associated digital scholarly objects for the purpose of facilitating the digital preservation of the evidence-base necessary for future understanding, evaluation, and reproducibility of scientific claims.

10 Things

News

Data Quality Review

CURE Training

Get Involved!

10 Things for Curating Reproducible and FAIR Research

Computational reproducibility requires a village. This document is primarily for data curators and information professionals who are charged with verifying that a computation can be executed and can reproduce prespecified results. Secondly, it is for researchers, publishers, editors, reviewers, and others who have a stake in creating, using, sharing, publishing, or preserving reproducible research.

The 10 Things for Curating Reproducible and FAIR Research is the result of the collaborative efforts of members of the Research Data Alliance (RDA) CURE-FAIR Working Group. The original 10 Things document was accepted by RDA as an endorsed recommendation cited below:

Arguillas, F., Christian, T., Gooch, M., Honeyman, T., & Peer, L. (2022). *10 Things for Curating Reproducible and FAIR Research* (Version 1.1). Research Data Alliance. <https://doi.org/10.15497/RDA00074>

What about workflows?



- Scientific computational workflows (written in e.g. Snakemake or NextFlow) may be shared just like any kind of code. However, it is often better to follow specific guidelines for sharing workflows.
- The organisations' behind the workflow management systems typically maintain their own documentation for how to share/publish workflows.
- You may publish your workflow in generic repositories like [Zenodo](#) or Figshare (e.g. [SciLifeLab Data Repository](#)) but WorkflowHub (<https://workflowhub.eu>) is probably a better place.





A software management plan should **minimally** include:

- What is expected to be produced (incl. documentation)?
- Who is responsible for releasing the software?
- What revision control process to be used?
- What license(s) will be used?



Adapted from <https://www.software.ac.uk/resources/guides/software-management-plans>

Software management plan \neq system management plan



A software management plan **could also**:

- identify the software development model to be used
- identify the external software that will be brought into the project, and the associated licences
- what method will be used to accept each component being produced
- dependencies between developed components and with external dependencies
- major risks that might impact the delivery



Adapted from <https://www.software.ac.uk/resources/guides/software-management-plans>

What organizations can do



- Identify code and software within the organization
- Evaluate the sustainability of the organization's software (e.g. using Software Sustainability Institute's [online evaluation tool](#))
- Create software management plans where needed
- Create a software and code sustainability plan for the whole organization
- Establish policies and best practices
 - Where to store and maintain code
 - How is code documented?
 - How should releases be named?
 - What code should be published?
 - Where should code be published?



What would you like to change regarding how your organization manages code and software?

Thanks for your attention!



Do you remember the presentation's key message?