FAIR & Research Software

Dr. Anna-Lena Lamprecht (@al_lamprecht)
Utrecht University, Netherlands

Dr. Salvador Capella-Gutierrez (@sj_capella)
Barcelona Supercomputing Center, Spain

CSDMS Webinar, 6 May 2021
Outline

Recap: The FAIR data principles
Data and software
Towards FAIR principles for research software
Outcomes so far
FAIR4RS community and ways to get involved
Are you familiar with the FAIR data principles?

- Yes: 11
- Somewhat: 3
- No: 1
"(Meta)data are assigned a globally unique and persistent identifier." Which FAIR principle?
"(Meta)data meet domain-relevant community standards." Which FAIR principle?
"(Meta)data are retrievable by their identifier using a standardised communications protocol." Which FAIR principle?
"(Meta)data include qualified references to other (meta)data." Which FAIR principle?
Leaderboard

3235 p | Agent X9
2763 p | me (fastest)
2758 p | Holahop
2729 p | Leslie
2614 p | Croc
2548 p | Ribbit
2536 p | Dr Centipede
2383 p | Greg
2294 p | St Elmo
1925 p | Snuggles
2016: "The FAIR guiding principles for scientific data management and stewardship" (Wilkinson et al., doi:10.1038/sdata.2016.18)

2018: "Central to the realization of FAIR are FAIR Digital Objects, which may represent data, software or other research resources." (European Commission)

But: How do the FAIR Principles relate to software?
Mismatch between the broad intentions of the 4 foundational FAIR principles and how the 15 FAIR Guiding Principles are communicated and perceived.
An ongoing discussion...


Most viewed article in Data Science in the first half of 2020!
Let's get into some of the thoughts and ideas in there.
<table>
<thead>
<tr>
<th>Recorded information</th>
<th>observations/measurements</th>
<th>Observations collected by humans or sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information in its raw-est form</td>
<td>Measured values of phenomena</td>
<td>information in a communicable form</td>
</tr>
<tr>
<td>Information in numerical or other forms</td>
<td>Recorded information</td>
<td></td>
</tr>
</tbody>
</table>
What is software?

- Computer code written to accomplish a purpose
- Code packages to perform some operation
- Tools based on a computer used to perform specific tasks, perhaps involving the use/output of data, but not necessarily

- Code applied to data
- Tool built to support a process, sometimes data collection
- Scientific software: a set of equations, captured in a language to mimic a process

- Computer code composed in a human-readable programming language
- Methodology
- Computer program structured to receive digital information and perform tasks
What is software?

- source code and its resultant executable application
- source code that required computational dependencies
What do data and software have in common?

- Software is data
- Both are stored on a computer
- They support research outputs and are very valuable
- Data have input and output that related to software. Software is computational engine to use data

- Part of scientific process
- Both are forms of information; like computer code, most data today exist in digital form.

- Used together to solve a problem
- Must have standard to work together
- Both can have multiple versions
What do data and software have in common?

- use interpreted language to communicate
- The need to be FAIR
What makes data and software different?

- **Software without data aren't too useful (for science)**
- **how they accumulate**
- **data is understandable using metadata. Software can't understand using metadata itself**

- **Data are not executable**
- **Software can be re-run many times, data observations aren't repeated exactly**
- **they require different knowledge to create, implement, document**

- **Software needs data to work**
- **data doesn't change and software can be versioned**
- **data is independent of a function, execution, an operable structure**
What makes data and software different?

- different expertise is needed to create each
- Data represent, whereas software instructs
- Ideally - provide use cases for when not to make it open?
Software is (not) data
Software is (not) data
Research software is "software that is used to generate, process or analyze results that you intend to appear in a publication" (Hettrick et al., 2014).

Many forms.
Many purposes.
Many distribution channels.

Traditionally, often created as Free and/or Open Source Software (FOSS).
Clear overlap of objectives, but not the same.

**FOSS**: Open source code, open licenses.

**FAIR**: Open data not a requirement.
Due to, e.g., privacy and sensitivity concerns with patients' health records.
Not in the same way valid for research software.
There is even a demand to make methods available!

Should FAIR software require FOSS?
Ongoing discussion ...
What do you think? Should FAIR require software to have an open license? Why, or why not?

Yes, on the principle of Accessibility

In some cases would be a benefit.

both are integral parts of the Open Science, Open Data, Open Source

No, because sometimes you are constrained to use some specific software that is not open

Yes, it is the default.

Should contain a strong recommendation towards an open license. At least have an executable open available (but code only recommended)

Open license by default, which exceptions allowable with good justification

Some commercial software is very useful for science - so perhaps providing instructions to repeat what you did that someone with a license should be able to execute

no, there are multiple ways to execute a program, requirement would lead to community adoption of a program as the only method, and limit progress
What do you think? Should FAIR require software to have an open license? Why, or why not?

No, quality assessment is a subjective issue
Software quality is a major concern in RSE. Can FAIR meet the expectations?

Distinguish between **form** and **function** of software:

Quality of the **form** of software can be covered by FAIR (code quality, maintainability).

Quality of the **functionality** of software goes beyond FAIR (functional correctness, software security, computational efficiency).
Should FAIR also take content quality into account? Why (not)?

There is no "Q" in FAIR...

No, because different scientific fields have different standards for accuracy/quality so blanket statements are difficult.

Quality is included implicitly, e.g. in R-reusable software should have undergone code review to ensure "correctness".

No - I'm not a computer scientist, but the code I write works for my analyses even if it isn't pretty.

Ahh, I don't know.... No?

No, that's a different issue.

Quality may be described as part of a metadata.

No, perhaps just structural conformance that helps domain specific reviewers assess quality more easily.

Tangential, but knowing your code is used by others may improve its quality.
RDA FAIR for Research Software (FAIR4RS) WG
https://www.rd-alliance.org/groups/fair-research-software-fair4rs-wg

Jointly convened as an RDA Working Group, FORCE11 Working Group, and Research Software Alliance (ReSA) Taskforce.

First draft of FAIR Principles for Research Software presented on 21 April 2021, see https://www.researchsoft.org/news/2021-04/
Draft FAIR Principles for Research Software

Findable: The software, and its associated metadata, should be easy to find for both humans and computers.

F1. Software is assigned a globally unique and persistent identifier that supports assigning of versions

F2. Software is described with rich metadata to support search and discoverability

F3. Metadata clearly and explicitly include the identifier of the software they describe

F4. Software is registered or indexed in a searchable resource

Accessible: The software, and its metadata, must be retrievable via standardized protocols.

A1. Software is retrievable by its identifier using a standardized communications protocol
   - A1.1. The protocol is open, free, and universally implementable
   - A1.2. The protocol allows for an authentication and authorization procedure, where necessary

A2. Metadata are accessible, even when the software is no longer available

Interoperable: The software interoperates with other software through exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs).

I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards

I2. Software includes qualified references to other objects

Reusable: The software is both usable (it can be executed) and reusable (it can be understood, modified, built upon, or incorporated into other software).

R1. Software is richly described with a plurality of accurate and relevant attributes
   - R1.1. Software is made available with a clear and accessible license
   - R1.2. Software is associated with detailed provenance

R2. Software includes qualified references to other software

R3. Software meets domain-relevant community standards

research data sharing without barriers
rd-alliance.org

FORCE11  ReSA  RDA

The Future of Research Communications and e-Scholarship
Research Software Alliance
Research Data Alliance
Another ongoing discussion.
How to measure software FAIRness?
Introduction: OpenEBench Technical Monitoring

At OpenEBench, we aim to provide and monitor **technical details** and **quality indicators** of bioinformatic tools.

- For **individual tools** to aid selection by scientists along with benchmarking results. 
  [OpenEBench technical monitoring](#)
- For the **whole population** of tools. 
  [OpenEBench Tools Observatory](#)
Introduction: Software Quality Monitoring

**Quality assessment framework**

- Towards FAIR principles for research software
- Software Management Plan.
- Workshops at ELIXIR All Hands.
- Community-led discussions by RDA, FORCE11 and ELIXIR on how to effectively apply FAIR principles to research software.

**Infrastructure**

- ETL strategies
- Metrics generation tools
- Platform to release results
Data Extraction and Transformation: Overview

Various sources

Tools metadata acquisition

FAIR metrics generation

Analysis and Visualization

We have a set of metrics following a FAIR schema and are progressively implementing them.
Results: Licensing

In some cases the licenses obtained from different sources are not coherent:

- Three different licenses: 0.0341%
- Two different licenses one of which is not recognized: 0.443%
- Two different recognized license: 0.409%

Licensing is one of the most crucial features of a piece of software, determining both its **Accessibility** and **Reusability**

Among the **unambiguous** licenses we find:

- None: 2.15%
- Other: 26.1%
- Open Source: 71.8%

If a license is NOT stated, the software CANNOT be used!!
Results: Licensing

Main Open Source License “families” distribution

- GPL: 5000 instances
- MIT: 2000 instances
- artistic: 1500 instances
- BSD: 1000 instances
- LGPL: 500 instances
- apache: 500 instances
- AGPL: 100 instances
- CC: 100 instances
- CeCILL: 50 instances

Open Source 71.8%
Other 26.1%
None 2.15%
Version control offers a standardized record of source code changes, making it easier to be Reused. The main version control systems also allow straightforward Accessibility.
"Five Recommendations for FAIR Software",
Netherlands eScience Center, https://fair-software.eu/

1. Use a publicly accessible repository with version control.
2. Add a license.
3. Register your code in a community registry.
4. Enable citation of the software.
5. Use a software quality checklist.

(Your organization can endorse this!)
How about your software? Do you...

- Use a publicly accessible repository with version control? 6.4
- Give it a licence? 6
- Register your software in a community registry? 3.3
- Enable citation of your software? 5.1
- Use a software quality checklist? 3.3


"FAIR Computational Workflows", Goble et al., Data Intelligence, 2020, https://doi.org/10.1162/dint_a_00033

"From FAIR research data toward FAIR and open research software", Hasselbring et al., Information Technology, 2020, https://doi.org/10.1515/itit-2019-0040

To the numerous people who contributed to the discussions around FAIR research software at different occasions and keep the work going!
Thank you!