

The FAIR Principles for Data and Software

Liam Pattinson

- 1 What are the FAIR Principles?
- 2 Motivation
- 3 Implementing the FAIR Principles
- 4 Conclusions

What are the FAIR Principles?

What are the FAIR Principles?

- The FAIR principles are a set of guidelines for managing (meta)data/software in a research setting, with the goal of encouraging open science, data/software reuse, and reproducibility.
- They were first formalised in an influential paper, “The FAIR Guiding Principles for scientific data management and stewardship”, *Scientific Data*, 3(1), 1-9, 2016, <https://doi.org/10.1038/sdata.2016.18>.
- They were later adapted for scientific software in “Introducing the FAIR Principles for research software”, *Scientific Data*, 9(1), 622, 2022, <https://doi.org/10.1038/s41597-022-01710-x>.
- A ‘living document’ can be found at <https://www.go-fair.org/>.

The FAIR Principles

F
Findable



A
Accessible



I
Interoperable



R
Reusable



FAIR Principles, (CC BY-SA 4.0)

The FAIR Principles

In order to be considered FAIR, data and software must be:

- **Findable:** (Meta)data and software should be easily discoverable by humans and machines.
- **Accessible:** Once (meta)data/software has been found, there should be a standard method of retrieving it.
- **Interoperable:** After accessing (meta)data/software, it should be possible to integrate it into wider workflows.
- **Reusable:** (Meta)data/software should be licensed appropriately, and sufficiently complete/well-described to enable reuse/replication.

We'll explore each of these concepts in greater detail later.

Motivation

Motivation

What are the problems the FAIR principles are intended to solve?

Reproducibility

- Reproducibility is at the heart of the scientific method. However, it is often the case that modern research results prove impossible to reproduce – the ‘replication crisis’.
- This is sometimes only revealed years after a study is published, and the conclusions may have already been adopted by the wider scientific community.
- Example, meta-studies found only 11% of ‘landmark’ studies developing new oncology drugs could be confirmed¹.
- Fraudulent research can have disastrous effects (e.g. the MMR vaccine scandal).
- It is much easier to expose non-reproducible research if the methodology and data are shared openly.

¹“Raise standards for preclinical cancer research”, Begley & Ellis, *Nature*, 483, 531-533, 2012, <https://doi.org/10.1038/483531a>.

Motivation

Efficiency

- A lot of research effort is redundant. Group A might spend a lot of time and money developing software or gathering data to solve a specific problem, unaware that group B has already done so (Findability).
- Even if a group is aware of similar work that has been done before, it might not be usable in their workflows (Interoperability), or might be strictly licensed so that they're not allowed to use it (Reusability).
- Data or software that is findable, accessible, interoperable, and openly licensed may still be unusable by other researchers if it is poorly described (Reusability).

Motivation

Impact

- Your work is more likely to be utilised by others if the associated data and software is available.
- Papers indicating available data are cited 25% more across disciplines².
- By following the FAIR principles, you can publish citable 'research outputs' with or without an associated paper.

²"Data sharing practices and data availability upon request differ across scientific disciplines", Tedersoo et al., Sci Data 8, 192 (2021), <https://doi.org/10.1038/s41597-021-00981-0>.

Motivation

Open scholarship

- The moral argument: publicly funded research should be publicly available (where appropriate). Taxpayers enable the research, but researchers and private journals block access to it.
- The lack of open access also impedes scientific progress.
- Easy to blame the large journals, but individual researchers also contribute to this research culture. Data requests to authors are successful in just 27-59% of cases, depending on the field³.
- **NOTE:** Data/software must be FAIR in order to be truly open, but it does not need to be open to be FAIR!

³“Data sharing practices and data availability upon request differ across scientific disciplines”, Tedersoo et al., Sci Data 8, 192 (2021), <https://doi.org/10.1038/s41597-021-00981-0>.

Motivation

Funding requirements

- Most UKRI funded projects now require a data management plan.
- EPSRC (engineering/physics) require data to be retained for 10 years after publication of a related paper, and to be as open as is reasonable.
- Following the FAIR principles will fulfil most requirements.

Implementing the FAIR Principles

The FAIR Principles – Findable

- Data and software can only be reused if people can find it.
- Ideally, it should be possible for somebody to find your data/software without needing to read an associated paper first.
- Sometimes data is kept private for good reason (personal information, government secrets, commercial interests, data embargo, etc). However, it is still possible to openly publish metadata while restricting full access only to authorised parties.

The FAIR Principles – Findable

- You'll need to select a *searchable* service to host your data. Some domain-specific repositories and databases exist:
 - [Climate Data Store](#)
 - [Protein Data Bank](#)
- [Zenodo](#) is a great general data resource.



The FAIR Principles – Findable

Software hosting is more complicated:

- Source code should be hosted using a version control service such as [GitHub](#), [GitLab](#), or [BitBucket](#).
- Software can also be uploaded to Zenodo. GitHub and Zenodo accounts can be linked to automatically keep software uploads in sync.
- Additionally, installable packages/binaries can be hosted on language-specific repositories.
 - Python packages should be uploaded to [PyPI](#), or perhaps [conda-forge](#).
 - R packages are managed using [CRAN](#).
 - Rust uses [crates.io](#).
 - Julia is unusual in that it uses GitHub as an external repository, but it is managed using the Pkg package manager.

The FAIR Principles – Findable

F1: (Meta)data/Software are assigned a globally unique and persistent identifier

- Each version of a dataset can be assigned a Digital Object Identifier (DOI). These can be assigned to any research output, from papers to raw data. E.g. <https://doi.org/10.5281/zenodo.7268985>
- DOIs persist even when your research outputs are moved.
- Software should be regularly ‘released’ with version numbers. Each version should have its own DOI and descriptive metadata (a *changelog*). This can be automated by linking GitHub and Zenodo.
- Components of the software representing levels of granularity are assigned distinct identifiers, e.g. plug-ins, 2D algorithm vs 3D algorithm.
- Authors can have their own persistent identifier via **ORCID**: <https://orcid.org/0000-0002-9079-593X>
- Cross-reference everything!

The FAIR Principles – Findable

F2: Data/Software are described with rich metadata

- Metadata is simply ‘data that describes data’. Having good metadata is essential if others are to find and understand your data/software.
- There are several main types of metadata:
 - **Administrative:** The stuff relevant for the management of data, which exists before a dataset is even created. E.g. What project is it for? Who funded it? Who owns it?
 - **Descriptive:** What is the purpose of the dataset? Why/how/when was the data gathered? Essential to aid the discovery of your data. Includes title, abstract, authors, keywords.
 - **Structural:** Information describing the data itself. How is the data structured? What columns/groups/variables does it contain? What are their units?
 - **Legal:** Licensing information, describing the terms of (re)use.
- There is often some overlap regarding which category metadata falls into, e.g. legal or descriptive metadata might also be considered administrative.
- A dataset has ‘rich metadata’ if it contains as much metadata as possible in all of the above categories.

The FAIR Principles – Findable

F3: Metadata clearly and explicitly include the identifier of the data/software they describe

- Sometimes the metadata and the dataset are located in separate files/repositories. The metadata should contain the DOI of the dataset to ensure the association is explicit.
- You can reserve DOIs on Zenodo, so you can add them to your README files before uploading.

F4: (Meta)data/software are registered or indexed in a searchable resource

- You can meet all of the above criteria, but your data/software may still be unfindable via most search engines if it's not stored using the correct service.
- Zenodo is a good general-purpose searchable resource, but your data may be more likely to be found in a domain-specific repository.

The FAIR Principles – Accessible

- ‘Accessibility’ is closely linked to ‘Findability’: there should be a clear path to access the data after finding it.
- This is about *technical* accessibility, not *legal* accessibility – that will be covered under ‘Reusable’.
- Encourages open data sharing, but does not mandate it.
- Almost identical between the FAIR data principles and FAIR software principles.

The FAIR Principles – Accessible

A1: (Meta)data are retrievable by their identifier using a standardised communications protocol

- The protocol should be open, free, and universally implementable, such as HTTP, FTP, etc. This just means people can access your data using a web browser – not a proprietary/specialised tool.
- If there are data privacy concerns, there should be a system in place to authorise access to specific users who meet the required criteria.
- If your preferred data hosting service doesn't allow fine-grained access control, stating 'email the authors for access' might be as FAIR as you can get.
- Zenodo allows protected datasets, for which individuals must request access and prove they meet the required criteria.

The FAIR Principles – Accessible

A2: Metadata should be accessible even when the data is no longer available

- It isn't practical to store data forever due to the cost of long-term storage. Access rights might also expire (e.g. GDPR).
- The metadata might remain useful anyway, as it lets researchers know what has been studied previously and by whom.
- Also permits data embargos – the institute generating data can release metadata instantly, but have privileged access for enough time to perform their own analyses. Again, this can be managed with Zenodo.

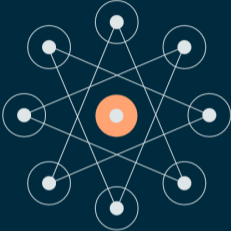
The FAIR Principles – Accessible

UKAEA Open Data

Published Data MAST Data License Get in touch

MAST Data

These pages provide an access point to publicly funded MAST research data. By selecting a Program or Objective, and then a specific experiment number, you can request the related underlying shot data if it is available for release.



- Example: The [MAST data set](#). This contains data from plasma physics experiments using the Mega Ampere Spherical Tokamak.
- Can search for specific ‘shots’, and search by metadata **[F]**.
- Published data is downloadable **[A]**.
- Need to manually request access to non-published data **[~A]**.
- The data is presented in unusual file formats, which causes issues with interoperability. . .

The FAIR Principles – Interoperable

- Data will often need to be combined with other data. Software will need to interact with other software.
- About *technical* interoperability, not *legal* interoperability.
- Can't predict who in the future might want to use our data/software, or what their workflow might be.
- Best to rely upon domain-relevant community standards, and to avoid creating new standards where possible.



The FAIR Principles – Interoperable Data

I1: (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation

- The data format should be formally specified and open.
- The format should be designed to be used in more than one scenario.
- Use file types that everyone else can read. No need to reinvent the wheel:
 - Use TOML/YAML for input files (although YAML *technically* isn't formalised).
 - .csv/HDF5/NetCDF4 for output files.
 - Simple .txt or markdown for README files.
 - Documentation can be provided via HTML or .pdf.
- Some controversy over whether .docx or .xlsx files are 'interoperable'. I'd argue that they aren't!

The FAIR Principles – Interoperable Data

I2: (Meta)data use vocabularies that follow the FAIR principles

- Is there a standard way to represent some (meta)data? If so, use it!
- E.g. standardised chemical naming conventions.
- For standard (meta)data vocabularies, see <https://fairsharing.org>, or the [RDA Metadata Standards Catalog](#).
 - The Climate and Forecast (CF) metadata conventions
 - Recommended Metadata for Biological Images (REMBI)

The FAIR Principles – Interoperable Software

I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.

- Making software interoperable can be much harder than making data interoperable, depending on how the user is expected to interact with the software.
- If software is *executable*, e.g. a script or compilable to .exe, the way it interacts with other software is via data IO. If you follow the FAIR data principles with your executables, you automatically get software interoperability!

The FAIR Principles – Interoperable Software

I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.

- Making *libraries* interoperable is more difficult, as the data exchange occurs at the Application Programming Interface (API). It's up to you to ensure that the entry point to your software is compatible with other related software.
 - Very important to write good documentation, especially for the public-facing API.
 - *Encapsulation*: expose only the necessary functions/classes in your API, keep the internal implementation private.
 - Use data types that are near-universally adopted where possible. E.g. in Python, use the 'Scientific Python Ecosystem': NumPy, SciPy, Pandas, Matplotlib, etc.
 - Follow best practices for 'packaging' – this controls how others can install your software and integrate it into their own workflows.

The FAIR Principles – Interoperable

I3: (Meta)data/software include qualified references to other (meta)data/software

- Applies to both data and software.
- A 'qualified reference' is a cross-reference that signals intent:
 - Dataset A *builds upon* dataset B.
 - Dataset X *is referenced by* paper Y.
 - Software library I *depends upon* library J.
 - Software P *is supplemental to* dataset Q.
- Reference by DOI where possible.

The FAIR Principles – Reusable

- ‘Reusability’ covers several subtopics:
 - Legal accessibility/interoperability: Even if users can find and access your data/software, and they’re able to fit it into their workflow, they may still be restricted from using it due to licensing restrictions.
 - Quality metadata: Data/software can only be reused if it can be understood.
 - Standards compliance: It’s easier to reuse data/software if it’s similar to other data/software already in use.
- The FAIR principles technically don’t include more abstract ideas of data/software quality, but they also contribute to reusability.

The FAIR Principles – Reusable

R1: (Meta)data/Software are richly described with a plurality of accurate and relevant attributes

- F2 details how metadata should be used to make data/software *discoverable*. This details how to make it *usable*.
- Should describe not just *what* the data/software is (structural metadata), but also the context in which it was created: *how, why, when*, and by *whom* (descriptive metadata). This lets other users know if it is actually relevant for their own studies.
- For data: which experimental protocols were used? Which machine was used? What software processed it? What was the scope of the original research?
- Software requires rich documentation. Describe which algorithms are used, include examples of proper usage, document potential pitfalls, etc.
- Be clear about the limitations of your data/software.
- You don't know what others might find useful. Include as much (meta)data as possible, even if it doesn't seem immediately relevant. Raw data may be more valuable to others than processed.

The FAIR Principles – Reusable

R1.1: (Meta)data/software are released with a clear and accessible usage license

- Beyond supplying structural and descriptive metadata, we also need to include *legal* metadata.
- Data and software should have clear licenses for reuse:
 - [Creative Commons](#) usually used for open data.
 - The [MIT License](#) software license is very permissive and easily understandable.
 - [GNU GPL v3.0](#) is a copyleft software license. Distributions of your software *must* include the source code, and software that includes⁴ GPL-licensed components *must* also be GPL. Encourages open source/open science, but can be too restrictive for some libraries.
- If access is restricted, make it clear what conditions are needed for access/reuse.

⁴Includes linking against GPL components for compiled software, and importing GPL Python code. Be wary of this license in your own dependencies!

The FAIR Principles – Reusable

R1.2: (Meta)data/software are associated with detailed provenance

- Provenance: the origin and/or history of your data/software.
- Who generated this data set? When? What other data sources did it rely on? How was the data processed?
 - Some software tools can generate this info for you in a machine-readable format ([Fair Data Pipeline](#), [AiiDA](#)).
- For software, maintain changelogs for each release.

The FAIR Principles – Reusable

R1.3: (Meta)data/software meet domain-relevant community standards

- Similar to I2: If there is a standard way to present (meta)data in your field, such as a particular database schema or vocabulary, use it.
- This may not be formalised in your field, but you should try your best to work in a way which minimises barriers for others.
- For software, follow universal language standards where possible. Don't rely on vendor-specific compiler extensions. It can also help to stay behind the bleeding edge.
- Your language choice also matters. R might be a good choice in bioinformatics, but not so much in physics. Similarly, Fortran is commonly used in physics, but not in most other domains.
- If other software in your domain prefers some specific file/data type, your software should do so too. For example, if it's standard for similar Python libraries to use NumPy arrays to pass data around, yours should too (even if, internally, it handles data differently).

The FAIR Principles – Reusable

R2: Software includes qualified references to other software.

- List your dependencies clearly, with bounds placed on their versions.
 - Overly strict dependency bounds can limit interoperability/reusability. Overly loose bounds can result in unstable software.
- Also list optional dependencies, with descriptions of what additional features are gained by including them. For example, some might be used just for running tests or building docs, but others might add additional features to your software.
- Via proper packaging methods (e.g. `pyproject.toml` in Python, modern Cmake for C/C++/Fortran), you can automate the process of getting the right versions of each dependency.

The FAIR Principles – Reusable

As stated earlier, the FAIR principles don't make any explicit statement on the quality of data or software. However, high quality data/software is much more likely to be reused in practice.

Extra: Data quality

- Provide enough (meta)data so that somebody else working in your domain can fully understand and replicate your results.
 - This might mean providing raw data alongside processed data.
- Data should be extendable, and applicable to more than a single problem.
- If possible, include errors associated with your data.

The FAIR Principles – Reusable

Software quality is difficult to measure. Here are some general tips to keep in mind when writing code:

Extra: Software quality

- Software should be, in order of importance:
 - 1 Correct
 - 2 Maintainable
 - 3 Optimised
- Operate on the 'principle of least surprise'. The functions you expose in your API should do simple, straightforward things with their inputs, with a minimum of side-effects.
- The KISS principle: 'Keep It Simple, Stupid!'. A convoluted solution to a problem might become unmaintainable in future.
- A solid test suite is a good marker of quality software!
- Documentation should be extensive, and not just comments in code. Use language-specific documentation generators (Sphinx, Doxygen, etc.).

Closing Thoughts on the FAIR Principles

- FAIRness is a spectrum: you don't need to tick every box for your data/software to be considered 'FAIR'. The intention is to improve the ways in which we manage and share data/software, and there's always room for further improvement.
- Try to consider machine actionability where possible. If your data/software is findable and retrievable by purely automated means (e.g. data in a domain-specific standard database, software in a package repository), it's much more likely to be reused.

Conclusions

Conclusions

- The FAIR principles for data/software are guidelines to help make your data/software more reusable.
- This improves the impact of your work, encourages open science, and makes research more reproducible.
- Data and software are treated similarly in most aspects, but there are some major differences:
 - Software can be hosted on general data repositories such as Zenodo, but it should also be hosted on dedicated services such as GitHub.
 - Software interoperability depends on how the software fits into existing workflows. Libraries require careful API design.
 - Data is fixed upon release, but software may have dependencies which change over time. Making software reusable means choosing dependencies carefully and setting appropriate version bounds.

Further Reading



- University of York Research Data Management guide at <https://subjectguides.york.ac.uk/rdm>.
- This includes material on the FAIR data principles <https://subjectguides.york.ac.uk/rdm/FAIR>.
- Living document for the FAIR principles: <https://www.go-fair.org/>.
- Carpentries course on FAIR data in biosciences: <https://carpentries-incubator.github.io/fair-bio-practice/>