Making Research Software FAIR: Background and Practical Steps
Date: 7 May 2025
Presented by: Bhavesh Patel (FAIR Data Innovations Hub, California Medical Innovations Institute)

(The slides are available via the link in the page's sidebar.)

Q: Can you talk more about how your group interprets “accessible” for closed-source software?

A: We interpret “accessible” as intended in the original FAIR Principles and the FAIR4RS Principles, i.e., the access process to a software should be clear, meaning that the software should be accessible through a standard protocol (such as over the internet) and that the access process (who can access, how, etc.) should be clearly described. Accessible here does not mean accessible to anyone (FAIR is not equal to open).

Q: Does the Codefair automation update, e.g. CITATION.cff *after* a DOI has already been created based on a GitHub tag? Is there any way to update citations and incorporate it into a tag, or do we always have to do that after? PS, a Release is based on a tag. A tag is just a tag. That’s how it works, right?

A: Codefair creates a draft release on Zenodo which allows to reverse a DOI (the DOI is not registered yet by Zenodo), then the DOI is added to the CITATION.cff and codemeta.json files, a GitHub release is created, that release’s zip is added to the Zenodo draft, information from the the CITATION.cff and codemeta.json files is used to automatically populate the Zenodo metadata, and finally the Zenodo release is made (Zenodo registers the DOI at that point).

Q: You mentioned the importance of updating the README file. What content do you think should be included in a README file?

A: To align with the FAIR4RS principles, we recommend to include the following elements in the README as part of the FAIR-BioRS guidelines:

- Overall description of the software (e.g., in an “About” section)
- High-level dependencies of the software (e.g., Node or Python version)

- Inputs and outputs of the software, parameters and data required to run the software
- The standards followed
- How to contribute to the software
- How to cite the software

This may evolve as part of the work of the Actionable FAIR4RS Task Force

Q: The FAIR principles for research software don't mention documentation to address reusability. At least, I don't see a principle related to documentation. This seems to be an omission to me. Is there anything about documentation in the FAIR-BioRS? I mean documentation of the interface for the software, like an API.

A: We do recommend having a README (and more for more complex software) as part of the FAIR-BioRS guidelines. I agree that a clear principle is missing in the FAIR4RS principles, something like "R4. Provide necessary documentation for reuse".

Q: Is FAIR optional? Is there any real consequence if it is not practiced?

A: FAIR is optional unless enforced by your funding source or your institution. The FAIR Principles and FAIR4RS Principles are really just things we should aspire to do with our data and software to do better science and enhance the pace of discoveries. Hopefully, with greater incentives, making data and software FAIR just becomes second nature and an integral part of scientific research just like how manuscript publication is.

Moderator's note: I have seen a few recent calls for proposals including requirements in their Data Management Plans for both the data and the software products to be made FAIR and for the proposers to explain how they will comply.