Implementing FAIR Biomedical Research Software

Bhavesh Patel, Ph.D.
Associate Research Professor

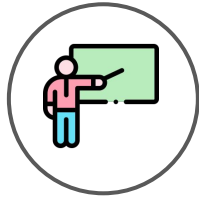CALIFORNIA MEDICAL INNOVATIONS INSTITUTE

FAIR DATA INNOVATIONS HUB

About This Presentation

15 min + 5 min Q&A

Learn why & how to make biomedical research software FAIR

About Research Software

About Research Software
Definition

**Any software created during the research process
or for a research purpose**

https://doi.org/10.5281/zenodo.5504016

About Research Software

There are many different types

Various formats

Python script

R code

Jupyter notebook

Various applications

Data analysis

Computational model

AI/ML model

About Research Software
It is an essential element of biomedical research...

More and more biomedical research projects include development of research software

Research software is the **main outcome of many research projects**

https://www.researchsoft.org/resa-resources

6

About Research Software
... making it FAIR is thus critical

Enable **reproducible, transparent research**

Prevent **duplicate effort**

Increase the **pace of discoveries** to improve human health

How to Make Research Software FAIR?

How to Make Research Software FAIR?
FAIR Principles

Findable, Accessible, Interoperable, and Reusable (FAIR) Principles (2016)

**Established to optimize the reusability
of all digital research objects, including software**

How to Make Research Software FAIR?
FAIR4RS Principles

FAIR Principles for Research Software (FAIR4RS Principles, 2022):
17 principles tailored for research software

F1. Software is assigned a globally unique and persistent identifier.
F1.1. Components of the software representing levels of granularity are assigned distinct identifiers.
F1.2. Different versions of the software are assigned distinct identifiers.
F2. Software is described with rich metadata.
F3. Metadata clearly and explicitly include the identifier of the software they describe.
F4. Metadata are FAIR, searchable and indexable.

(and more)

https://doi.org/10.1038/s41597-022-01710-x

How to Make Research Software FAIR?
Problem

The FAIR4RS Principles by design
do not provide actionable instructions

How do I assign a unique identifier?
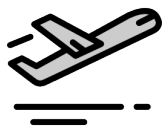
How do I provide rich metadata?

FAIR-BioRS Guidelines

FAIR-BioRS Guidelines
About

FAIR Biomedical Research Software (FAIR-BioRS) Guidelines

Minimal, actionable, step-by-step guidelines for
complying with each of the FAIR4RS principles

December 2021
Beginning of this effort

August 2023
Manuscript published

https://doi.org/10.1038/s41597-023-02463-x

FAIR-BioRS Guidelines
Development process

List of questions **+** Literature review **+** Our assessment **+** External feedback

https://doi.org/10.1038/s41597-023-02463-x

FAIR-BioRS Guidelines
Overview
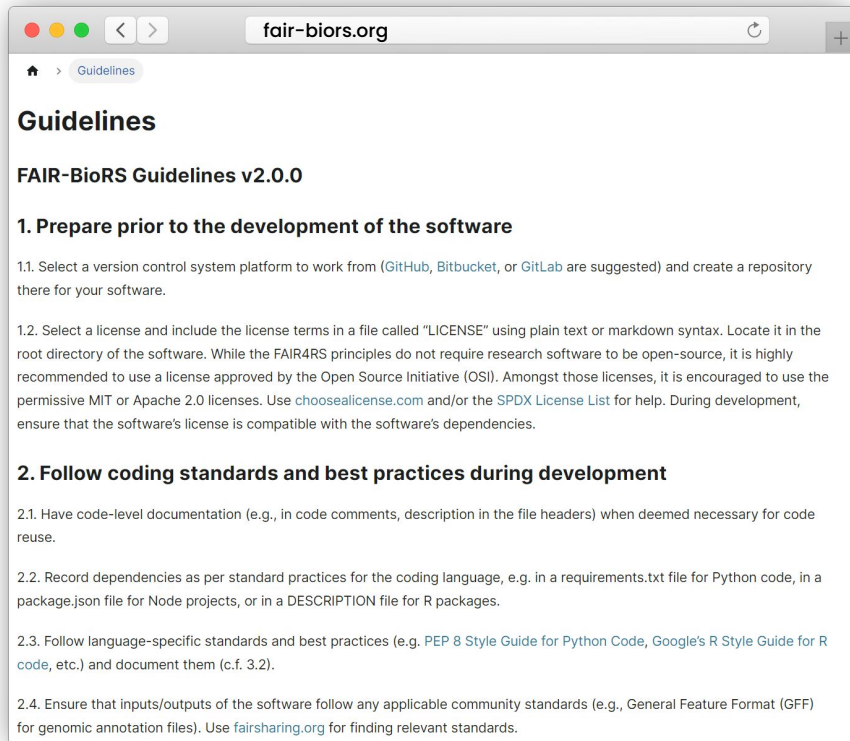
🔗 Full guidelines: fair-biors.org

📋 Step-by-step process

👆 Clear & actionable instructions

---

fair-biors.org

🏠 > Guidelines

Guidelines

FAIR-BioRS Guidelines v2.0.0

1. Prepare prior to the development of the software

1.1. Select a version control system platform to work from (GitHub, Bitbucket, or GitLab are suggested) and create a repository there for your software.

1.2. Select a license and include the license terms in a file called "LICENSE" using plain text or markdown syntax. Locate it in the root directory of the software. While the FAIR4RS principles do not require research software to be open-source, it is highly recommended to use a license approved by the Open Source Initiative (OSI). Amongst those licenses, it is encouraged to use the permissive MIT or Apache 2.0 licenses. Use choosealicense.com and/or the SPDX License List for help. During development, ensure that the software's license is compatible with the software's dependencies.

2. Follow coding standards and best practices during development

2.1. Have code-level documentation (e.g., in code comments, description in the file headers) when deemed necessary for code reuse.

2.2. Record dependencies as per standard practices for the coding language, e.g. in a requirements.txt file for Python code, in a package.json file for Node projects, or in a DESCRIPTION file for R packages.

2.3. Follow language-specific standards and best practices (e.g. PEP 8 Style Guide for Python Code, Google's R Style Guide for R code, etc.) and document them (c.f. 3.2).

2.4. Ensure that inputs/outputs of the software follow any applicable community standards (e.g., General Feature Format (GFF) for genomic annotation files). Use fairsharing.org for finding relevant standards.

FAIR-BioRS Guidelines
Overview

Metadata files

 codemeta.json

 CITATION.cff

4. Include metadata files #

4.1. Include a codemeta.json metadata file in the root directory of the software. The CodeMeta generator can be used. Provide, at least, the following fields:

- "name"
- "description"
- "identifier"
- "keywords"
- "programmingLanguage"
- "dataPublished"
- "dateModified"
- "license"
- "givenName", "familyName" and "affiliation" for each author.

When applicable, also provide the following fields: "isPartOf", "hasPart", and "relatedLink". See the CodeMeta documentation for a definition of the fields.

4.2. Include a CITATION.cff metadata file in the root directory of the software. The CFF file initializer can be used. Provide, at least, the following fields:
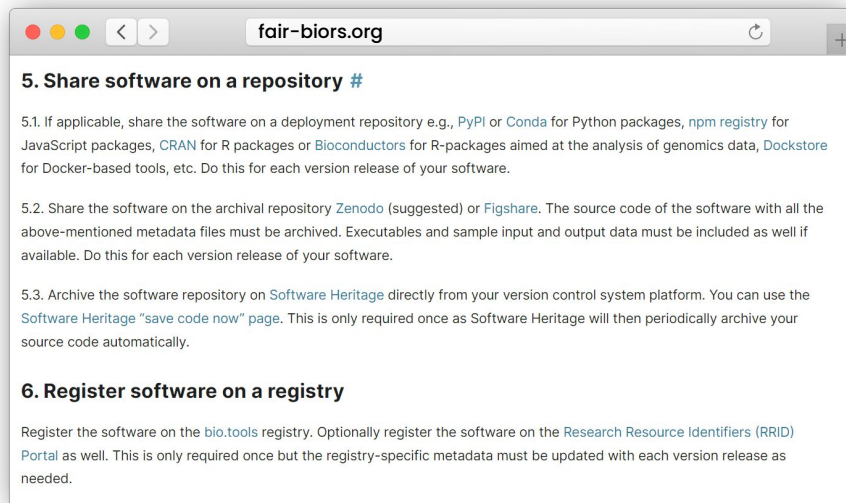
- "abstract"
- "identifiers"
- "keywords"
- "license"
- "date-released"
- "given-names", "family-names" and "affiliation" for each author.
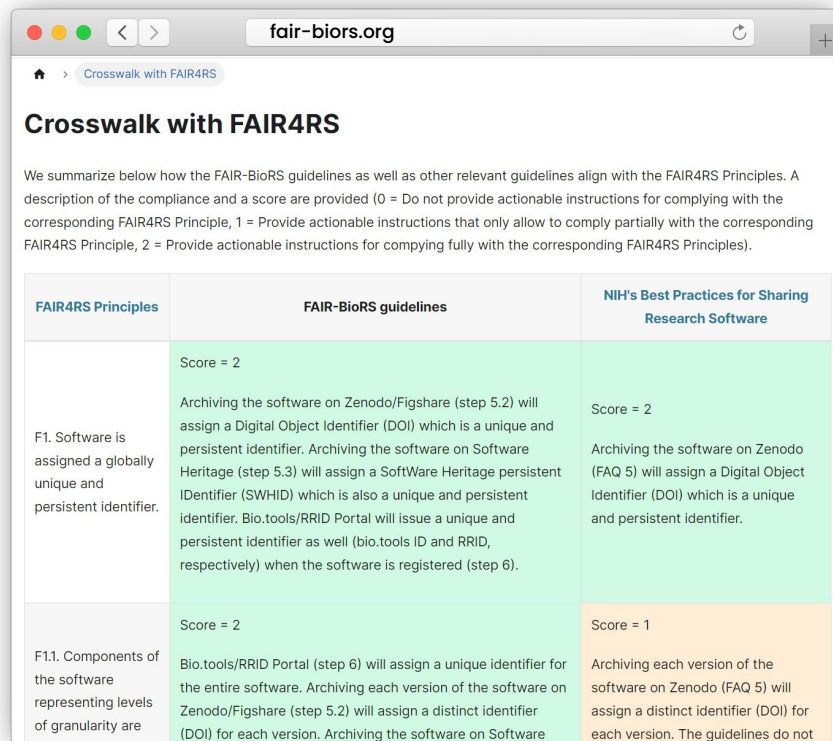
FAIR-BioRS Guidelines
Overview

Archival repositories (obtaining DOI)

5. Share software on a repository

5.1. If applicable, share the software on a deployment repository e.g., PyPI or Conda for Python packages, npm registry for JavaScript packages, CRAN for R packages or Bioconductors for R-packages aimed at the analysis of genomics data, Dockstore for Docker-based tools, etc. Do this for each version release of your software.

5.2. Share the software on the archival repository Zenodo (suggested) or Figshare. The source code of the software with all the above-mentioned metadata files must be archived. Executables and sample input and output data must be included as well if available. Do this for each version release of your software.

5.3. Archive the software repository on Software Heritage directly from your version control system platform. You can use the Software Heritage "save code now" page. This is only required once as Software Heritage will then periodically archive your source code automatically.

6. Register software on a registry

Register the software on the bio.tools registry. Optionally register the software on the Research Resource Identifiers (RRID) Portal as well. This is only required once but the registry-specific metadata must be updated with each version release as needed.

FAIR-BioRS Guidelines
Crosswalk with FAIR4RS Principles

Crosswalk with FAIR4RS

Crosswalk with FAIR4RS

We summarize below how the FAIR-BioRS guidelines as well as other relevant guidelines align with the FAIR4RS Principles. A description of the compliance and a score are provided (0 = Do not provide actionable instructions for complying with the corresponding FAIR4RS Principle, 1 = Provide actionable instructions that only allow to comply partially with the corresponding FAIR4RS Principle, 2 = Provide actionable instructions for compying fully with the corresponding FAIR4RS Principles).

FAIR4RS Principles	FAIR-BioRS guidelines	NIH's Best Practices for Sharing Research Software
F1. Software is assigned a globally unique and persistent identifier.	Score = 2 Archiving the software on Zenodo/Figshare (step 5.2) will assign a Digital Object Identifier (DOI) which is a unique and persistent identifier. Archiving the software on Software Heritage (step 5.3) will assign a SoftWare Heritage persistent IDentifier (SWHID) which is also a unique and persistent identifier. Bio.tools/RRID Portal will issue a unique and persistent identifier as well (bio.tools ID and RRID, respectively) when the software is registered (step 6).	Score = 2 Archiving the software on Zenodo (FAQ 5) will assign a Digital Object Identifier (DOI) which is a unique and persistent identifier.
F1.1. Components of the software representing levels of granularity are	Score = 2 Bio.tools/RRID Portal (step 6) will assign a unique identifier for the entire software. Archiving each version of the software on Zenodo/Figshare (step 5.2) will assign a distinct identifier (DOI) for each version. Archiving the software on Software	Score = 1 Archiving each version of the software on Zenodo (FAQ 5) will assign a distinct identifier (DOI) for each version. The guidelines do not

18

What's Next?

What's Next?

User-support tool: codefair (beta)

Open source and free GitHub app that acts as your personal assistant for making software FAIR

1
Install the app from GitHub Marketplace

2
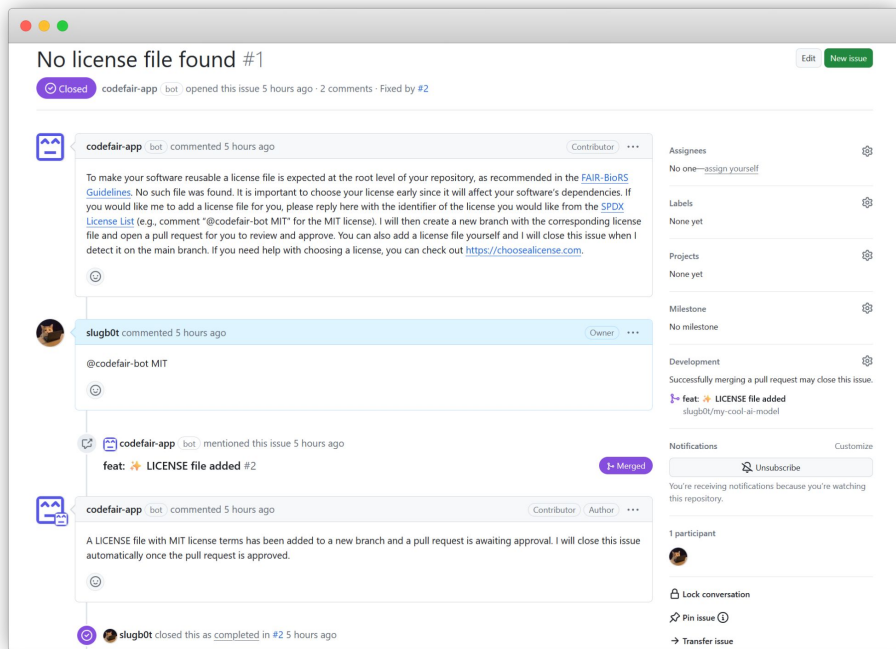Code and develop software as usual

3
Address GitHub issues and pull requests from the app

codefair.io

What's Next?

User-support tool: codefair (beta)

codefair.io

Closing Comments

Closing Comments
Main message

Research software is a critical element of biomedical research and needs to be included in the **discussions about FAIR**!

Thank You!

📧 **bpatel@calmi2.org**   🌐 **fairdataihub.org**

*Find these slides and
all resources here* ➡️

tinyurl.com/FAIRsoftwareBioIT