

Implementing FAIR Biomedical Research Software

Bhavesh Patel, Ph.D.
Associate Research Professor



CALIFORNIA MEDICAL
INNOVATIONS INSTITUTE



FAIR DATA
INNOVATIONS HUB



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>

About Research Software

... sharing and making it reusable is thus critical



Enable **reproducible, transparent research**



Prevent **duplicate effort**



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



How to Make Research Software Reusable?

How to Make Research Software Reusable?

FAIR Principles – Overview

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 Reusable?

FAIR4RS Principles – Overview

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)

How to Make Research Software Reusable?

FAIR4RS Principles – 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

FAIR-BioRS Guidelines

Development process



List of questions



Literature review



Our assessment



External feedback

FAIR-BioRS Guidelines

Overview



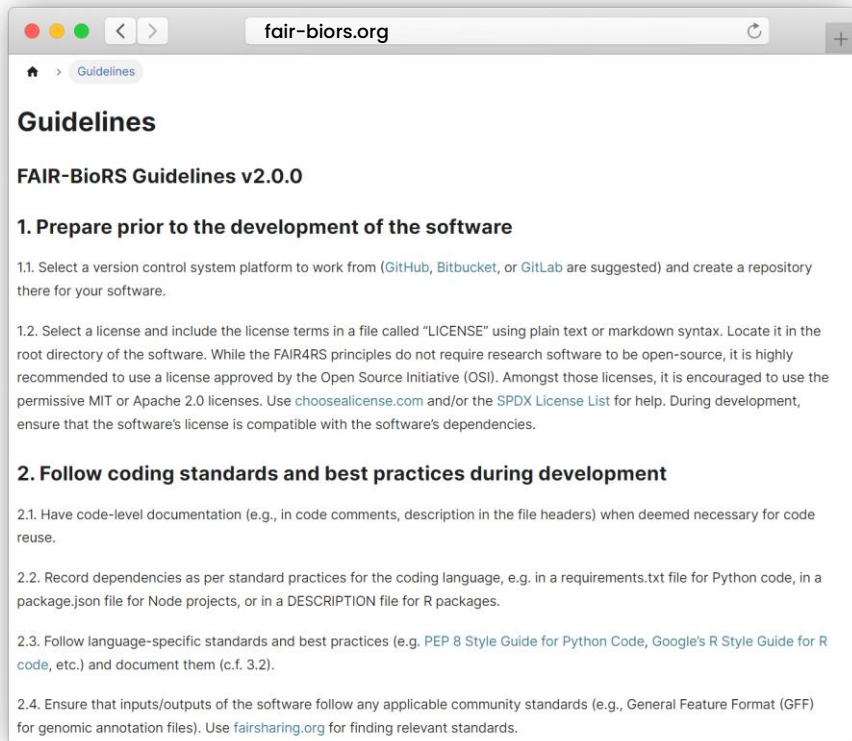
Full guidelines: fair-biors.org



Step-by-step process



Clear, actionable instructions

A screenshot of a web browser displaying the FAIR-BioRS Guidelines v2.0.0 page. The browser's address bar shows "fair-biors.org". The page has a header with a home icon and a "Guidelines" tab. The main content area is titled "Guidelines" and "FAIR-BioRS Guidelines v2.0.0". It lists two main sections: "1. Prepare prior to the development of the software" and "2. Follow coding standards and best practices during development". Section 1 includes two sub-points: 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. Section 2 includes four sub-points: 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

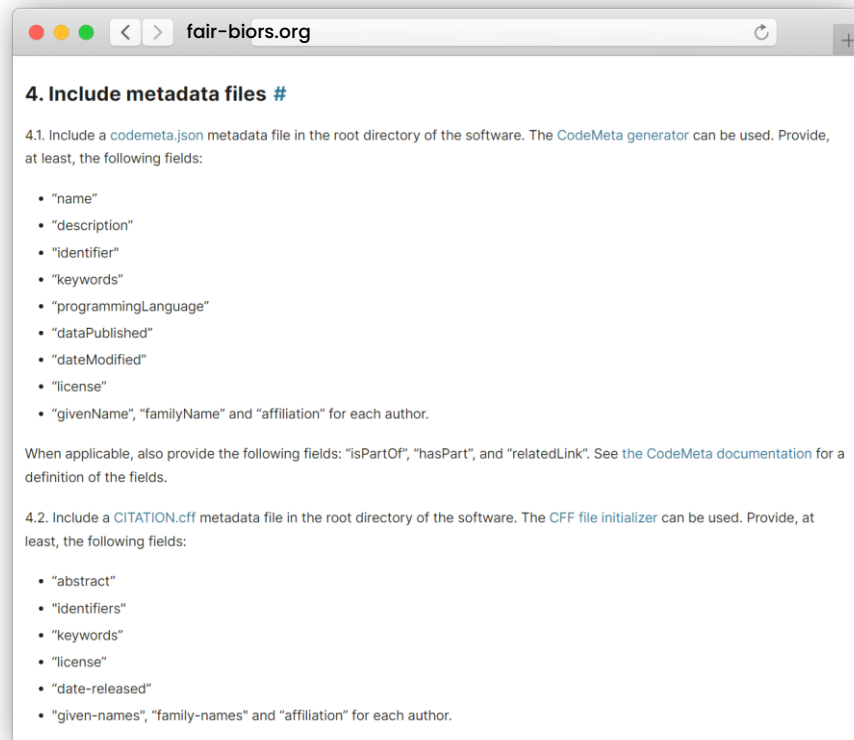
Metadata files



codemeta.json



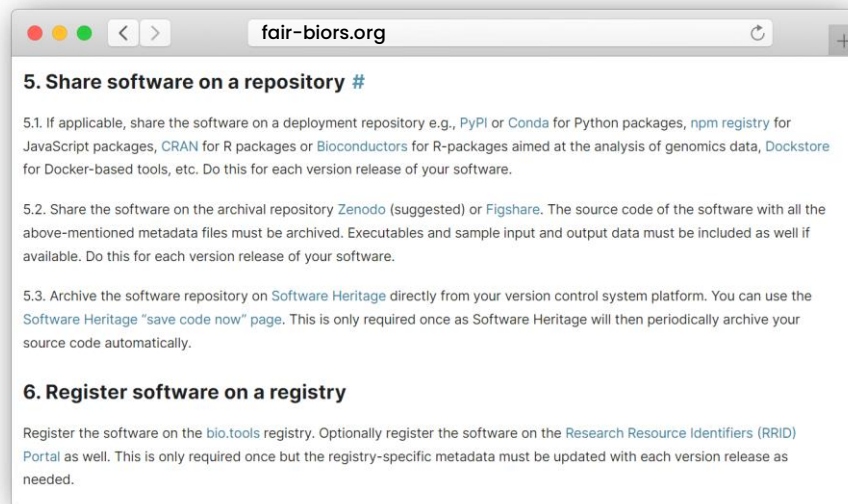
CITATION.cff



FAIR-BioRS Guidelines

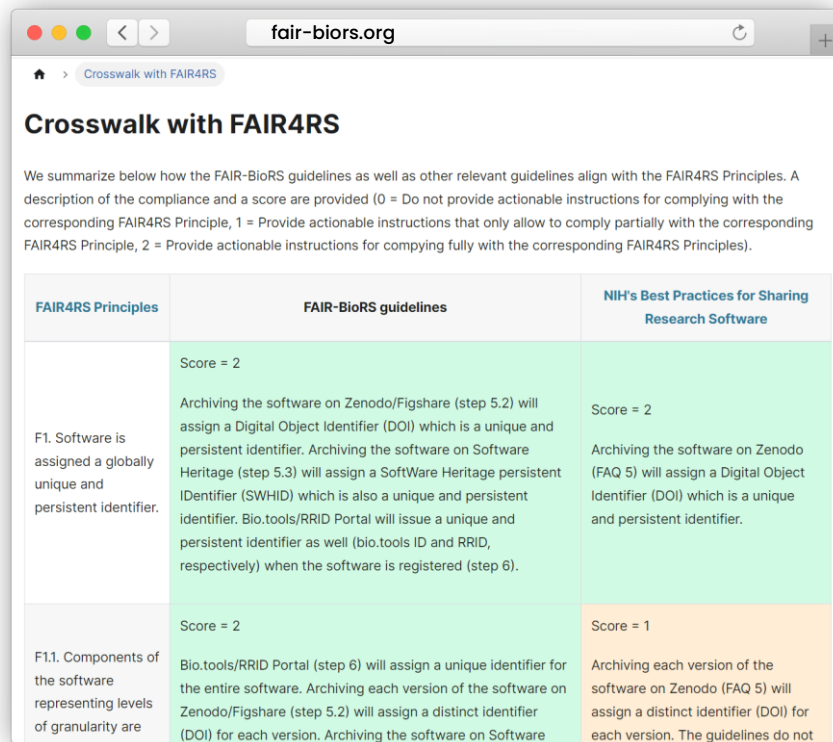
Overview

Archival repository (obtaining DOI)



FAIR-BioRS Guidelines

Crosswalk with FAIR4RS Principles



The screenshot shows a web browser window with the address bar displaying 'fair-biors.org'. The page title is 'Crosswalk with FAIR4RS'. Below the title, there is a paragraph explaining the purpose of the crosswalk: '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 complying fully with the corresponding FAIR4RS Principles).' Below this paragraph is a table with three columns: 'FAIR4RS Principles', 'FAIR-BioRS guidelines', and 'NIH's Best Practices for Sharing Research Software'.

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

FAIR-BioRS Guidelines

They benefit everyone!



For developers
of research software

Increase the impact of your
software

Get recognition for your
development effort



For funders

Increase your return on
investment

Avoid funding duplicate
development effort

Note: FAIR \neq Open



User-support Tools

User-support Tools

Tool #1: FAIRshare



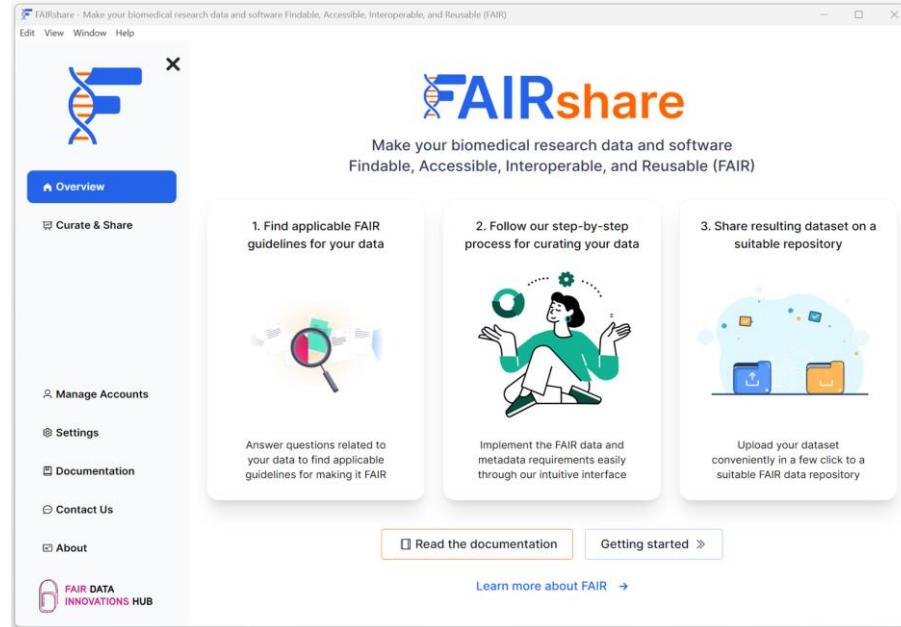
Cross-platform
desktop app



Open source



Free



User-support Tools

Tool #2: codefair.io (beta)



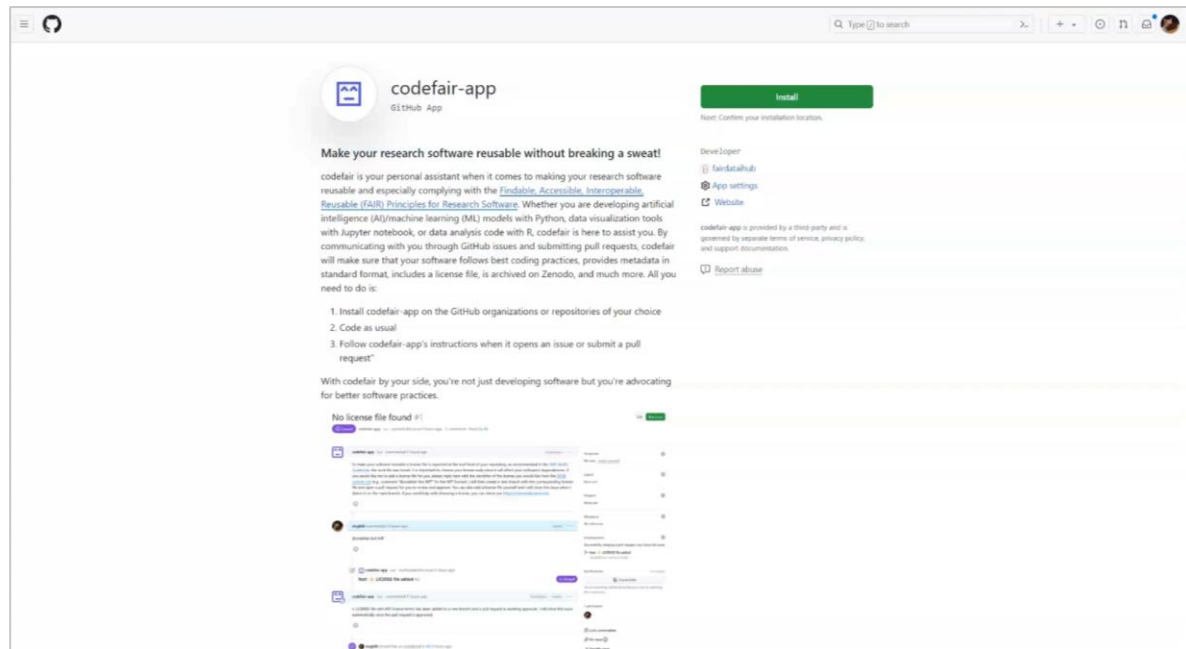
GitHub app



Open source



Free





Closing Comments

Closing Comments

How can you support this effort?



Follow the FAIR-BioRS guidelines (fair-biors.org)



Use support tools (docs.fairshareapp.io, codefair.io)



Reach out to us with questions (bpatel@calmi2.org)



Support the maintenance of the guidelines
and development of automation tools

Thank You!



bpatel@calmi2.org



fairdataihub.org

*Find these slides and
all resources here*



<https://tinyurl.com/fairsoftware>