Utrecht University

02-06-2022

Fair Metrics for FAIR Software

Dr. Anna-Lena Lamprecht, Utrecht University

NWO Open Science in Practice Series

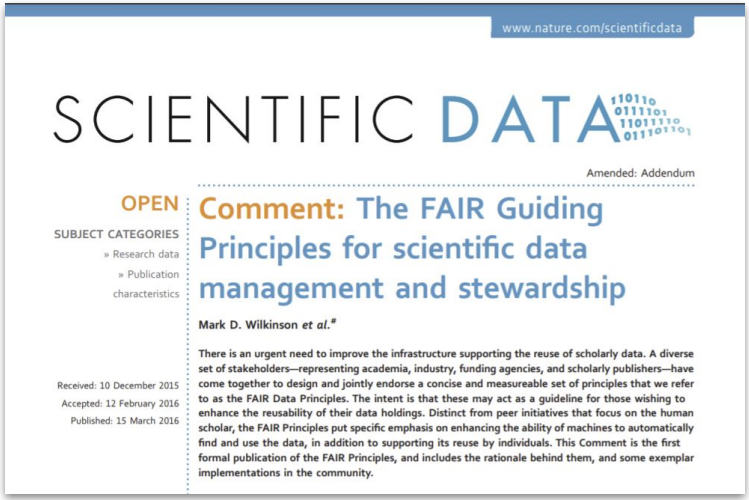
The FAIR Principles

Widely known as the “FAIR Data Principles”

Original paper demands that **all scholarly digital research objects** should be findable, accessible, interoperable and reusable

Increasingly recognized as essential for the transition towards Open Science

6700+ citations counted by Google Scholar



www.nature.com/scientificdata

SCIENTIFIC DATA

Amended: Addendum

OPEN

SUBJECT CATEGORIES

» Research data

» Publication characteristics

Received: 10 December 2015

Accepted: 12 February 2016

Published: 15 March 2016

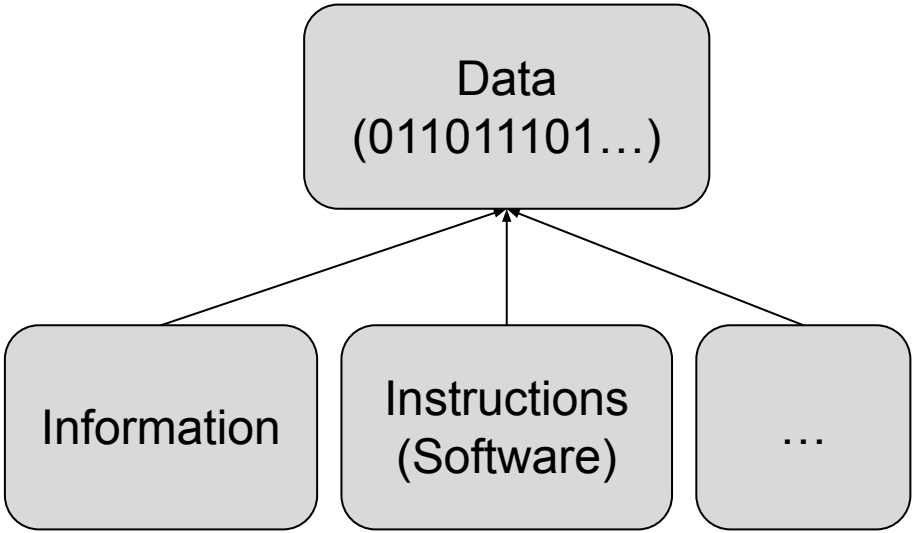
Comment: The FAIR Guiding Principles for scientific data management and stewardship

Mark D. Wilkinson *et al.*#

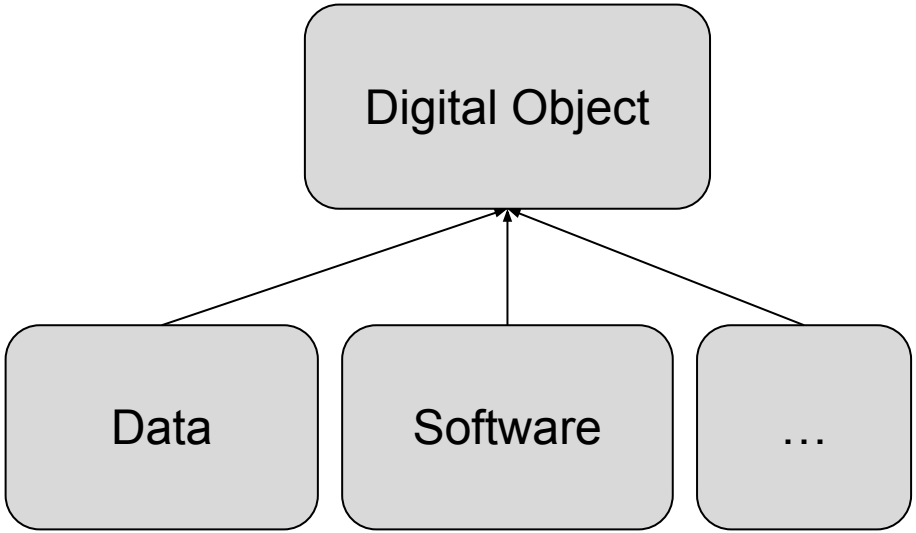
There is an urgent need to improve the infrastructure supporting the reuse of scholarly data. A diverse set of stakeholders—representing academia, industry, funding agencies, and scholarly publishers—have come together to design and jointly endorse a concise and measureable set of principles that we refer to as the FAIR Data Principles. The intent is that these may act as a guideline for those wishing to enhance the reusability of their data holdings. Distinct from peer initiatives that focus on the human scholar, the FAIR Principles put specific emphasis on enhancing the ability of machines to automatically find and use the data, in addition to supporting its reuse by individuals. This Comment is the first formal publication of the FAIR Principles, and includes the rationale behind them, and some exemplar implementations in the community.

Wilkinson, M., Dumontier, M., Aalbersberg, I. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data* 3, 160018 (2016). https://doi.org/10.1038/sdata.2016.18

Software is (not) Data



traditional computer science view



computational science view

More FAIR Stuff

FAIR code
FAIR workflows
FAIR ML models
FAIR microscopy images
FAIR teaching material
FAIR physical samples
FAIR hardware
FAIR batteries
...

FAIR4RS (FAIR for Research Software)

Workshop discussions

DTL Conference 2018 Utrecht, WOSSS 2019 The Hague, de-RSE 2019 Potsdam, Top 10 FAIR Things Global Sprint 2019, National eScience Symposium 2019 Amsterdam, ...

Lamprecht, Garcia, Kuzak, Martinez, et al.: **Towards FAIR Principles for Research Software.** Data Science, 2020, https://doi.org/10.3233/DS-190026.

International **FAIR4RS working grouped** formed (by RDA, ReSA & FORCE11), https://www.rd-alliance.org/groups/fair-research-software-fair4rs-wg

Five Recommendations for FAIR Software (https://fair-software.eu/)

Chue Hong, Katz, Barker, Lamprecht, Martinez, et al.: **FAIR Principles for Research Software (FAIR4RS Principles)**, RDA Recommendation, 2022, https://doi.org/10.15497/RDA00068

Key points of FAIR4RS

Software is not (just) data:

- Result of a creative process
- Executable
- Composite nature
- Frequent changes
- Decay

Warrant changes to several of the data-oriented principles.

Example principles:

F1. Software is assigned a globally unique and persistent identifier.

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

I2. Software includes qualified references to other objects.

R1. Software is described with a plurality of accurate and relevant attributes.

R2. Software includes qualified references to other software.

Fair Metrics for FAIR Software

Project No. 203.001.124
NWO Open Science Fund 2020/2021

Team:

- Anna-Lena Lamprecht (PI), Utrecht University
- Michelle Barker, Research Software Alliance
- Jonathan de Bruin, Utrecht University
- Carlos Martinez-Ortiz, Netherlands eScience Center
- Jurriaan Spaaks, Netherlands eScience Center
- Extra RSE
- Student assistants

Problem

The application of the FAIR principles to software lags significantly behind their application to data.

One reason: Current lack of evaluated, community-endorsed metrics to assess software FAIRness and the tooling to support such assessment.

They would enable users to choose the FAIRest option, and in turn incentivize developers to make their software FAIRer.

Vision

We envision actionable metrics and workable tools for assessing the FAIRness of software, which maximize the utility for research software developers and users, and incentivize cultural change in line with the ideas and ideals of Open Science.

Aims

Towards this vision, this project aims at evaluating suggested FAIR software metrics with a focus on how to incentivize cultural change.

Four main objectives:

1. Identify candidate metrics based on recent discussions in the community.
2. Prototypically implement tooling for applying the candidate metrics.
3. Evaluate the candidate metrics on software projects from different research areas.
4. Aggregate the findings into recommendations for FAIR software metrics and tooling.

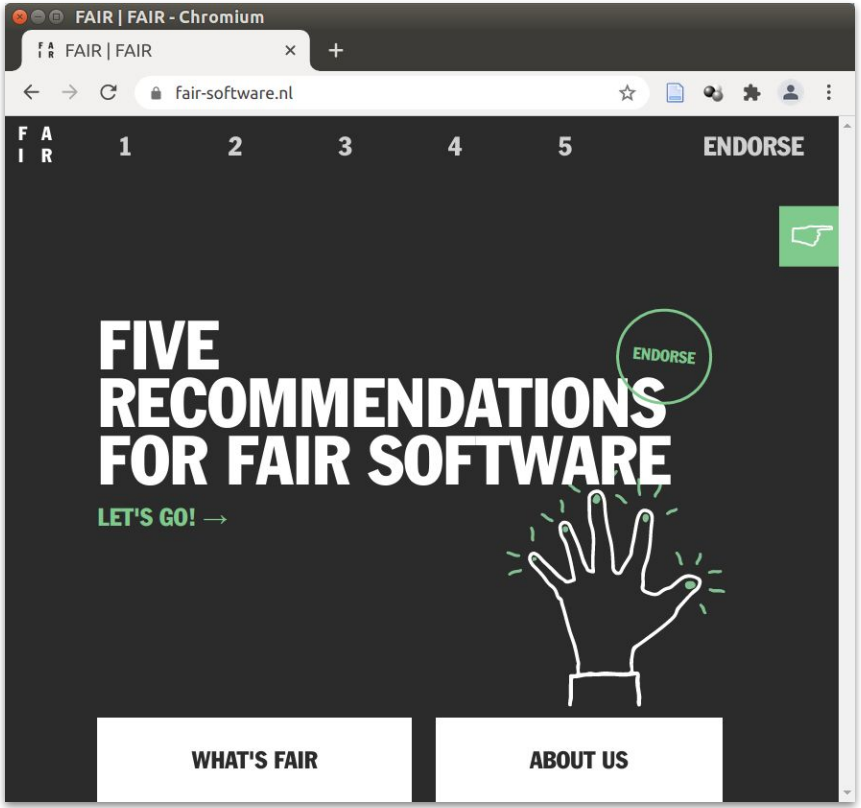
Candidate Metrics

- The Netherlands eScience Center's "Five Recommendations for FAIR software" (https://fair-software.nl/) and the howfairis tool (https://github.com/fair-software/howfairis)
- The Barcelona Supercomputing Center's "FAIRsoft approach": https://doi.org/10.1101/2022.05.04.490563
- Outcomes of the FAIR4RS Roadmap Metrics Working Group (https://docs.google.com/document/d/1BpzecVx4ZvSNfHD-UHhofZVdA6qiP_ENrmozmiq9zY4/edit)
- Australian Research Data Commons (ARDC) self assessment tool
- FAIR Software Assessment/Badging initiative (GO FAIR US)
- GO BUILD (FAIR assessment framework)

fair-software.nl (fair-software.eu)

Five Recommendations for FAIR Software (NLeSC, 2019):

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



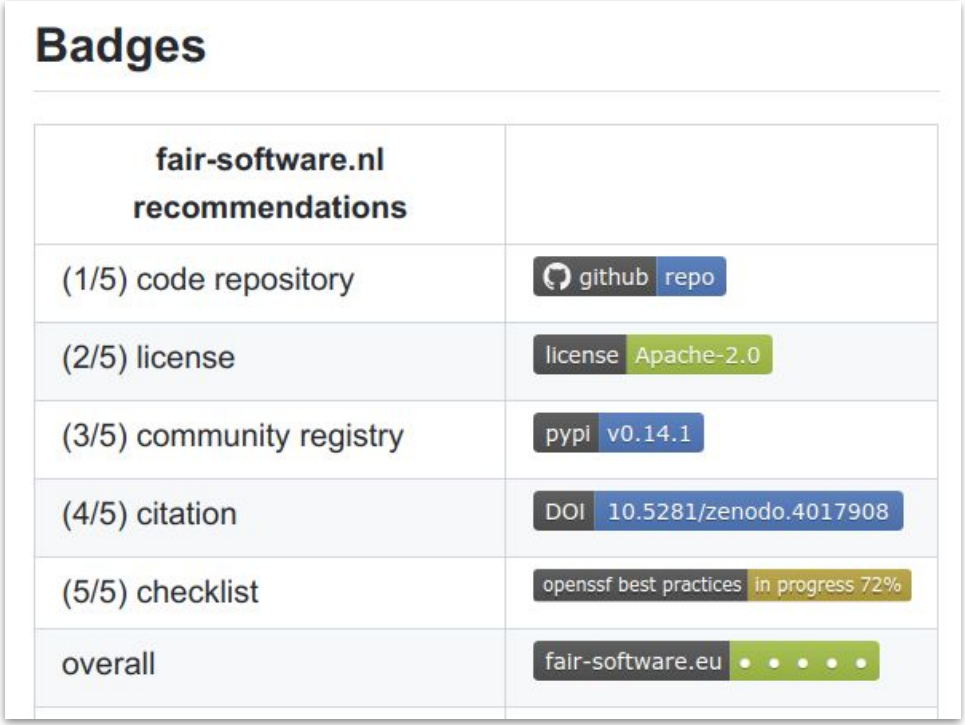
howfairis

https://pypi.org/project/howfairis/ or
https://github.com/fair-software/howfairis

Tool to analyze a GitHub/GitLab repository's compliance with the fair-software.nl recommendations (NLeSC, 2020)

Binary assessment of the five recommendations

Gives score (0-5) and badge



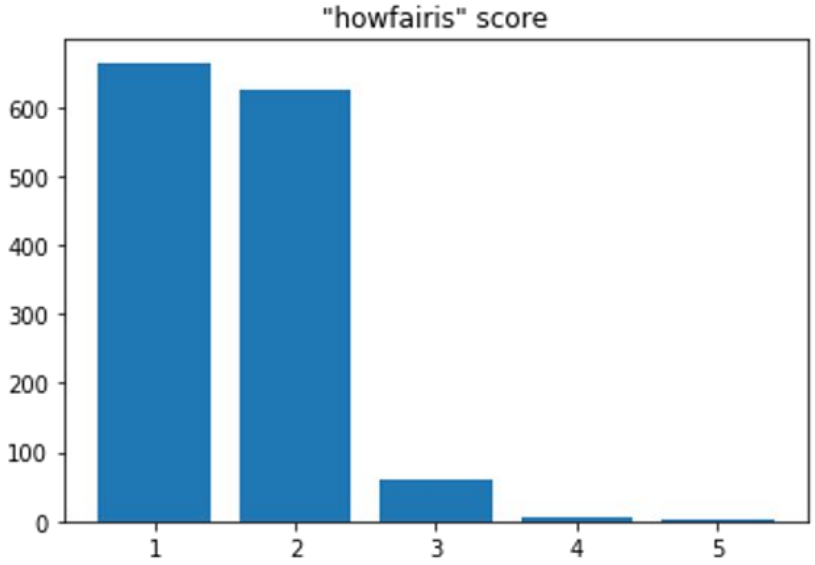
Screenshot from https://github.com/fair-software/howfairis

howfairis in SWORDS@UU

Scan and Review of Open Research Data and Software (SWORDS) @ UU (https://github.com/UtrechtUniversity/SWORDS-UU)

A project to get stats and figures about how Utrecht University researchers develop and manage software

Summer 2021: first results for 143 users and 1356 repositories associated with Utrecht University, incl. the howfairis score



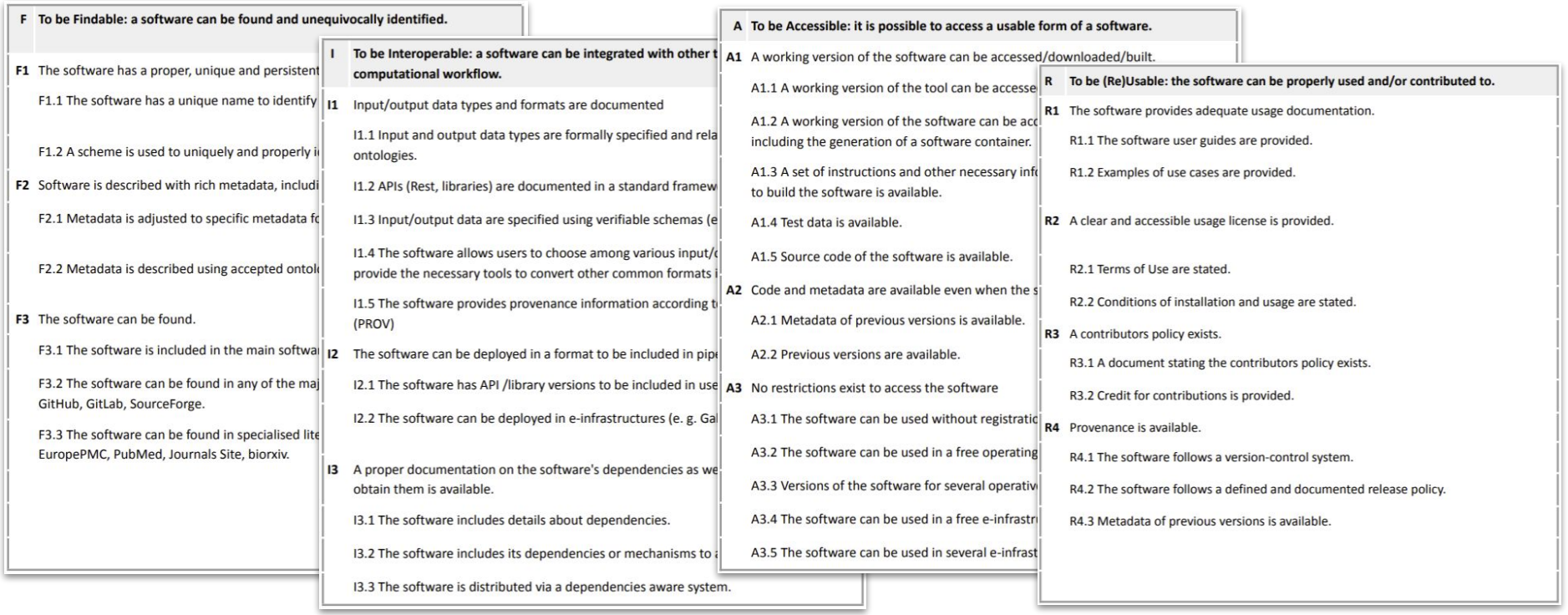
FAIRsoft

Eva Martín del Pico, Josep Lluís Gelpí, Salvador Capella-Gutiérrez:
FAIRsoft - A practical implementation of FAIR principles for research software
(bioRxiv preprint, https://doi.org/10.1101/2022.05.04.490563)

Key ideas:

- interpretation of the FAIR4RS principles as measurable indicators
- scoring system to weight the different indicators, to offer a quantitative assessment of software FAIRness

FAIRsoft: High-Level and Low-Level Indicators



F	To be Findable: a software can be found and unequivocally identified.
F1	The software has a proper, unique and persistent
	F1.1 The software has a unique name to identify
	F1.2 A scheme is used to uniquely and properly i
F2	Software is described with rich metadata, includi
	F2.1 Metadata is adjusted to specific metadata fo
	F2.2 Metadata is described using accepted ontol
F3	The software can be found.
	F3.1 The software is included in the main softwa
	F3.2 The software can be found in any of the maj GitHub, GitLab, SourceForge.
	F3.3 The software can be found in specialised lite EuropePMC, PubMed, Journals Site, biorxiv.

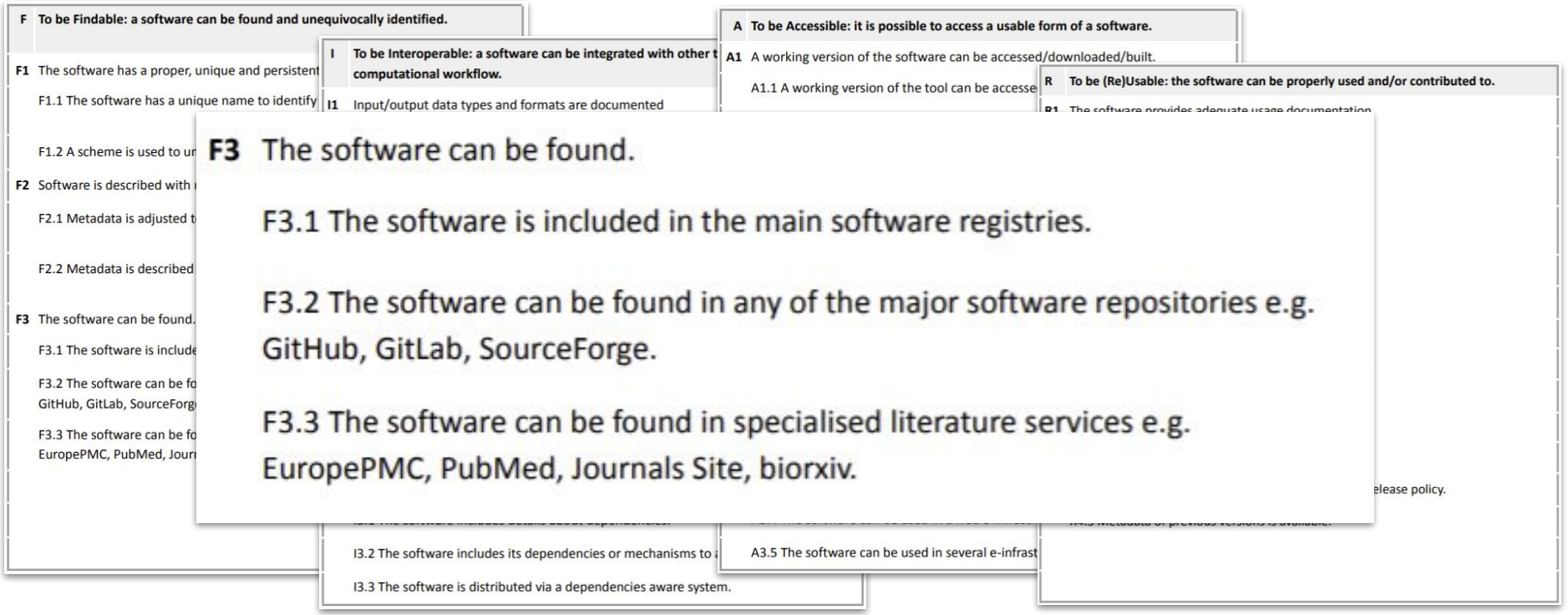
I	To be Interoperable: a software can be integrated with other t computational workflow.
I1	Input/output data types and formats are documented
	I1.1 Input and output data types are formally specified and rela ontologies.
	I1.2 APIs (Rest, libraries) are documented in a standard framew
	I1.3 Input/output data are specified using verifiable schemas (e
	I1.4 The software allows users to choose among various input/ provide the necessary tools to convert other common formats i
	I1.5 The software provides provenance information according t (PROV)
I2	The software can be deployed in a format to be included in pipe
	I2.1 The software has API /library versions to be included in use
	I2.2 The software can be deployed in e-infrastructures (e. g. Gal
I3	A proper documentation on the software's dependencies as we obtain them is available.
	I3.1 The software includes details about dependencies.
	I3.2 The software includes its dependencies or mechanisms to a
	I3.3 The software is distributed via a dependencies aware system.

A	To be Accessible: it is possible to access a usable form of a software.
A1	A working version of the software can be accessed/downloaded/built.
	A1.1 A working version of the tool can be accesse
	A1.2 A working version of the software can be acc including the generation of a software container.
	A1.3 A set of instructions and other necessary inf to build the software is available.
	A1.4 Test data is available.
	A1.5 Source code of the software is available.
A2	Code and metadata are available even when the s
	A2.1 Metadata of previous versions is available.
	A2.2 Previous versions are available.
A3	No restrictions exist to access the software
	A3.1 The software can be used without registratio
	A3.2 The software can be used in a free operating
	A3.3 Versions of the software for several operativ
	A3.4 The software can be used in a free e-infrastr
	A3.5 The software can be used in several e-infrast

R	To be (Re)Usable: the software can be properly used and/or contributed to.
R1	The software provides adequate usage documentation.
	R1.1 The software user guides are provided.
	R1.2 Examples of use cases are provided.
R2	A clear and accessible usage license is provided.
	R2.1 Terms of Use are stated.
	R2.2 Conditions of installation and usage are stated.
R3	A contributors policy exists.
	R3.1 A document stating the contributors policy exists.
	R3.2 Credit for contributions is provided.
R4	Provenance is available.
	R4.1 The software follows a version-control system.
	R4.2 The software follows a defined and documented release policy.
	R4.3 Metadata of previous versions is available.

From bioRxiv preprint doi: https://doi.org/10.1101/2022.05.04.490563

FAIRsoft: High-Level and Low-Level Indicators



From bioRxiv preprint doi: https://doi.org/10.1101/2022.05.04.490563

FAIRsoft: Scoring System

Low-level indicators are assigned weights according to their relative relevance for the respective high-level indicators

High-level indicators are assigned weights according to their relative relevance for the respective FAIR principle

Results in a score between 0 and 1 for each FAIR principle

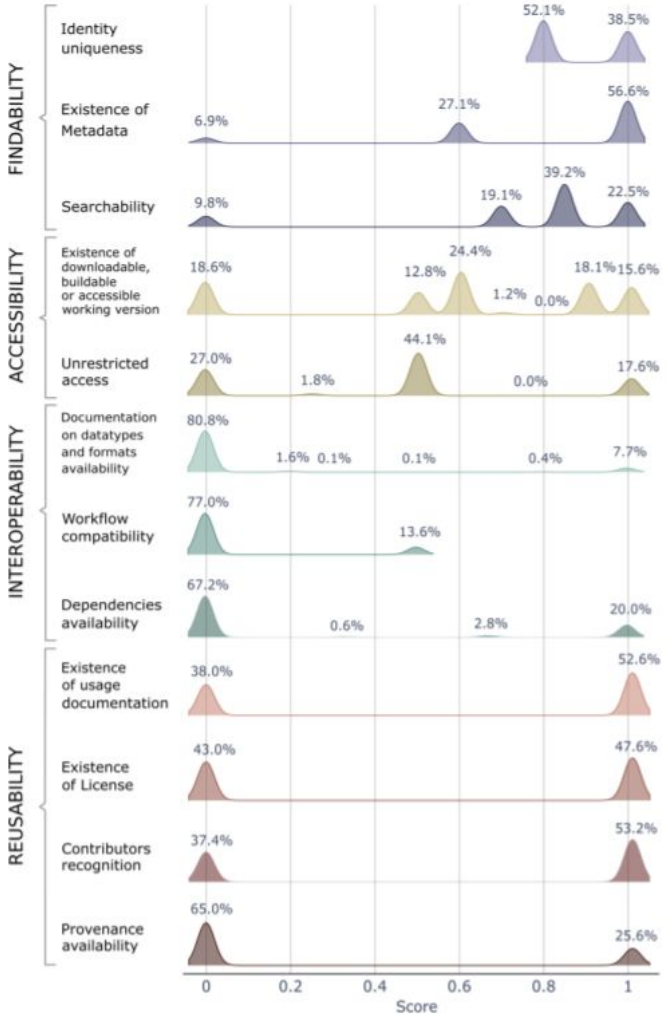
Identifier	Name	LL weight	HL weight
F1	**Identity uniqueness**		
F1.1	Uniqueness of name	0.8	0.4
F1.2	Identifiability of version	0.2	
F2	**Existence of Metadata**		
F2.1	Structured Metadata	0.6	0.2
F2.2	Standardized Metadata	0.4	
F3	**Searchability**		
F3.1	Searchability in registries	0.3	0.4
F3.2	Searchability in software repositories	0.3	
F3.3	Searchability in literature	0.4	

Example adapted from https://docs.google.com/document/d/1OcGxtX-LqloePSeEE-S1v3kgKXe6wiCgo_VMTF6idhM/edit#

FAIRsoft: Application to Life Science Tools

FAIRsoft indicators computed for 43,987 tool instances in the OpenEBench platform

Individual scores summarised to depict the FAIRness level of the evaluated collection



Screenshot from https://www.biorxiv.org/content/10.1101/2022.05.04.490563v1.full.pdf

Next in the Project

Finish review of suggested FAIR software metrics

Implement tooling for automatically accessible metrics

Apply to 100+ open source research software projects

Evaluation of the effects on their developers

Formulate recommendations for FAIR software metrics

Evaluation: Focus on Incentivization Potential

Goal are metrics that foster cultural change by incentivizing research software developers to make their software FAIR(er)

Points to consider:

- What are incentives for research software developers?
- Quantitative or qualitative FAIRness assessment?
- Score, badge or other form of summary?
- What is an appropriate level of detail?

Summary

FAIR essential for the transition towards Open Science

Software (and other kinds of research objects) need specialized FAIR principles

Various metrics for software FAIRness suggested

Project to evaluate metrics with focus on incentivization potential

Vision are metrics and tools that help fostering cultural change

Thank you!

Utrecht University

Sharing science,
shaping tomorrow