



# FAIR 4 Software

---

Sara El-Gebali

 [0000-0003-1378-5495](https://orcid.org/0000-0003-1378-5495)

 [@yalahowy](https://twitter.com/yalahowy)

# Housekeeping

Stay on mute!



Raise your /hand



Notes are for you



Code of conduct



Welcome!!



The image shows a HackMD document titled "FAIR4Software Workshop" with a rendered HTML view on the right. The code on the left includes a title, a welcome message, a code of conduct reminder, an agenda, and a rollcall prompt. The rendered HTML on the right displays these elements as a structured document with headings and lists.

```
1 # FAIR4Software Workshop
2 ## ==Slides available [here]==
3
4 ## Welcome 🙌
5
6 ## Code of Conduct reminder
7 * Be respectful, honest, inclusive, accommodating,
8 appreciative, and open to learning from everyone else.
9 * Do not attack, demean, disrupt, harass, or threaten others
10 or encourage such behavior.
11 * Be patient, allow others to speak, and use the zoom
12 reactions if you would like to voice something.
13
14 -----
15 # Agenda:
16 - Principles & definitions
17 - Application & implementation
18 - Challenges & Discussion
19
20 # Rollcall:
21 🗣️ Name / 🗨️ pronouns/ 🗨️ What are you hoping to get out of
22 this session?
23
24 *
25 *
26 *
27 -----
```

CHANGED 2 DAYS AGO

## FAIR4Software Workshop

**Slides available [here]**

Welcome 🙌

### Code of Conduct reminder

- Be respectful, honest, inclusive, accommodating, appreciative, and open to learning from everyone else.
- Do not attack, demean, disrupt, harass, or threaten others or encourage such behavior.
- Be patient, allow others to speak, and use the zoom reactions if you would like to voice something.

### Agenda:

- Principles & definitions
- Application & implementation
- Challenges & Discussion

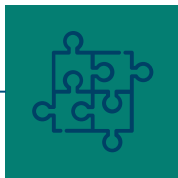
### Rollcall:

# Agenda



## Principles & Definitions

- What is FAIR?
- What is Data?
- What is Research Software?



## Application & Implementation

- FAIR 4 Software
- Implementation
- Why do it?



## Challenges & Discussion

- Current challenges
- Ongoing Work
- Open Discussions

# FAIR principles

FAIR is a set of principles to define the best practices for data to facilitate discovery, access and reuse by humans and machines.

FAIR is not rules and not a standard, it is an evolving process and a vision.

# FAIR principles

FAIR is a set of principles to define the best practices for data to facilitate discovery, access and reuse by humans and machines.

FAIR is not rules and not a standard, it is an evolving process and a vision.

What does **FAIR** stand for?

**F**indable, **A**ccessible, **I**nteroperable and **R**eusable.



# FAIR Data principles

[Home](#) / [Scientific data](#) / [Comment](#) / [Article](#)


[Open Access](#) | [Published: 15 March 2016](#)

## The FAIR Guiding Principles for scientific data management and stewardship

[Mark D. Wilkinson](#), [Michel Dumontier](#), [...] [Barend Mons](#) 

*Scientific Data* **3**, Article number: 160018 (2016) | [Cite this article](#)

**355k** Accesses | **2966** Citations | **1912** Altmetric | [Metrics](#)

 An [Addendum](#) to this article was published on 19 March 2019

### Abstract

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 measurable 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

# FAIR Data principles

[Home](#) / [Scientific data](#) / [Comment](#) / [Article](#)


[Open Access](#) | [Published: 15 March 2016](#)

## The FAIR Guiding Principles for scientific data management and stewardship

[Mark D. Wilkinson](#), [Michel Dumontier](#), [...] [Barend Mons](#) 

*Scientific Data* **3**, Article number: 160018 (2016) | [Cite this article](#)

**355k** Accesses | **2966** Citations | **1912** Altmetric | [Metrics](#)

 An [Addendum](#) to this article was published on 19 March 2019

### Abstract

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 measurable 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

# What is Data?

**Primary-** Raw from measurements or instruments

**Secondary-** Processed from secondary analysis and interpretations.

**Published-** final format available for use and reuse

**Metadata-** data about your data



# What is Data?

**Primary-** Raw from measurements or instruments

**Secondary-** Processed from secondary analysis and interpretation

**Published-** final format available for use and reuse

**Metadata-** data about your data



It is everything that you need to validate or reproduce your research findings as well as what is required for the understanding and handling of the data.

# What do we mean by Research Software?

In 2014, UK Research Software Survey chose to define Research software as:

“Software that is used to **generate, process** or **analyse** results intended for publication .... Research software can be anything from a few lines of code written by yourself, to a professionally developed software package.”

# What do we mean by Research Software?



## Defining Research Software: a controversial discussion

*Summary Report of FAIR4RS Subgroup 3 activity and discussion*

Morane Gruenpeter (Inria, Software Heritage), Daniel S. Katz (University of Illinois), Anna-Lena Lamprecht (Utrecht University), Tom Honeyman (Australian Research Data Commons), Daniel Garijo (Information Sciences Institute), Alexander Struck (Cluster of Excellence Matters of Activity, de-RSE), Anna Niehues (Radboud university medical center), Paula Andrea Martinez (Research Software Alliance), Leyla Jael Castro (ZB MED Information Centre for Life Sciences), Tovo Rabemanantsoa (French National Research Institute for Agriculture, Food and Environment), Esther Plomp (Delft University of Technology - Faculty of Applied Sciences), Neil Chue Hong, Carlos Martinez-Ortiz, Laurents Sesink, Matthias Liffers (Australian Research Data Commons), Anne Claire Fouilloux, Chris Erdmann, Silvio Peroni (University of Bologna), Paula Martinez Lavanchy, Ilian Todorov (UKRI)

[Defining Research Software: a controversial discussion](#) reviews existing definitions of research software in order to provide the overall context of the subgroup outputs ([Gruenpeter et al., 2021](#)).

# What do we mean by Research Software?



## Defining Research Software: a controversial discussion

*Summary Report of FAIR4RS Subgroup 2 activity and discussion*

### How to identify Research Software?

#### **Analysis of questions**

What can be considered as RS is difficult to agree upon, since usage of software is so abundant everywhere, including in research contexts. Should software used to write an article, e.g., Microsoft Word or LaTeX, be considered RS? The same can be asked of software used to capture data, which might also be used to analyze and process data, like Microsoft Excel: should this be identified as Research Software? Or do we exclude all

Morane Gruenpeter (Inria, Sof  
(Utrecht University), Tom Honey  
Institute), Alexander Struck (U  
university medical center), Paul  
Information Centre for Life Scienc  
Food and Environment), Esther F  
Hong, Carlos Martinez-Ortiz, Laur  
Fouilloux, Chris Erdmann, Silvio

[Defining Research Software: a controversial discussion](#) reviews existing definitions of research software in order to provide the overall context of the subgroup outputs ([Gruenpeter et al., 2021](#)).

# Is Software Data?

Let us know in the Chat!  
Yes/No/Maybe



# Is Research Software considered data?

***“ Software is data, but it is not just data.***

*While “data” in computing and information science can refer to anything that can be processed by a computer, software is a special kind of data that can be a creative, executable tool that operates on data.”*

[10.7287/peerj.preprints.2630v1](https://doi.org/10.7287/peerj.preprints.2630v1)

# Is Research Software considered data?

## 2. Software is not data

Technically, software is a special kind of data. In computing, digital data (ultimately sequences of ones and zeros) are used to represent all information, including factual data as well as computer instructions. In the more abstract context of FAIR, software and data are regarded as different kinds of digital research objects next to each other. As such, they share particular characteristics that allow them to be treated alike for certain aspects of FAIR, such as the possibility of having a Digital Object Identifier (DOI) assigned, or having a license. However, as elaborated by Katz et al. [4], there are also several significant differences between data and software as digital research objects: Data are facts or observations that provide evidence. In contrast, software is the result of a creative process that provides a tool for doing something, for example with data. As such, software is executable, while data is not. Software is often built using other software. This is especially obvious for software that implements multi-step processes to coordinate multiple tasks and their data dependencies, which are usually referred to as workflows [5,6]. Gener-

# Software vs. data in the context of citation

Daniel S. Katz<sup>1</sup>, Kyle E. Niemeyer<sup>2</sup>, Arfon M. Smith<sup>3</sup>, William L. Anderson<sup>4</sup>,  
Carl Boettiger<sup>5</sup>, Konrad Hinsén<sup>6</sup>, Rob Hooft<sup>7</sup>, Michael Hucka<sup>8</sup>, Allen Lee<sup>9</sup>,  
Frank Löffler<sup>10</sup>, Tom Pollard<sup>11</sup>, and Fernando Rios<sup>12</sup>

<sup>1</sup>National Center for Supercomputing Applications & Electrical and Computer  
Engineering Department & School of Information Sciences, University of Illinois

## LIST OF DIFFERENCES

### Software is executable, data is not

A commonsense definition of software is that it is “a set of instructions that direct a computer to do a specific task” (Chun, 2005). On the other hand, data is simply a collection of facts or measurements (real or simulated). In other words, software is functionally active, while data is passive. Of course, software (in form) can be considered data as well, especially to functional programmers familiar with





# **FAIR 4 Software (FAIR4S)**

# Findable

Software and associated metadata should be findable.

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

**F1.1:** Different components of the software must be assigned distinct identifiers representing different levels of granularity.

**F1.2:** Different versions of the same software must be 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 and are searchable and indexable.



# Accessible

Software and associated metadata must be retrievable via standardized protocols.

**A1:** Software is retrievable by its identifier using a standardized communications protocol.

**A1.1:** The protocol is open, free, and universally implementable.

**A1.2:** The protocol allows for an authentication and authorization procedure, where necessary.

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

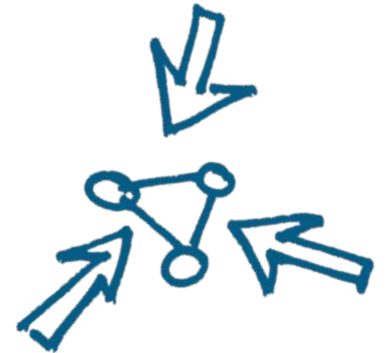


# Interoperable

The software interoperates with other software through exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs).

**I1:** Software reads, writes and exchanges data in a way that meets domain-relevant community standards.

**I2:** Software includes qualified references to other objects.



# Software Reuse vs Reproducibility

Software is both usable (it can be executed) and reusable (it can be understood, modified, built upon, or incorporated into other software).

<https://zenodo.org/record/5524726#.YYhA8tbMLt0>

# Software Reuse vs Reproducibility

Software is both usable (it can be executed) and reusable (it can be understood, modified, built upon, or incorporated into other software).

<https://zenodo.org/record/5524726#.YYhA8tbMLt0>

Software reproducibility here means the ability for someone to replicate a computational experiment that was done by someone else, using the same software and data, and then to be able to change part of it (the software and/or the data) to better understand the experiment and its bounds.

<://www.software.ac.uk/blog/2017-02-20-software-reproducibility-possible-and-practical>

# Reusable

The software is both usable (it can be executed) and reusable (it can be understood, modified, built upon, or incorporated into other software).

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

**R1.1:** Software must have a clear and accessible license.

**R1.2:** Software is associated with detailed provenance.

**R2:** Software includes qualified references to other software.

**R3:** Software meets domain-relevant community standards.



# Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions

*Fabien C. Y. Benureau*<sup>1,2,3\*</sup> and *Nicolas P. Rougier*<sup>1,2,3</sup>

<sup>1</sup>INRIA Bordeaux Sud-Ouest, Talence, France, <sup>2</sup>Institut des Maladies Neurodégénératives, Université de Bordeaux, Centre National de la Recherche Scientifique UMR 5293, Bordeaux, France, <sup>3</sup>LaBRI, Université de Bordeaux, Bordeaux INP, Centre National de la Recherche Scientifique UMR 5800, Talence, France

Scientific code is different from production software. Scientific code, by producing results that are then analyzed and interpreted, participates in the elaboration of scientific conclusions. This imposes specific constraints on the code that are often overlooked in practice. We articulate, with a small example, five characteristics that a scientific code in computational science should possess: re-runnable, repeatable, reproducible, reusable, and replicable. The code should be executable (re-runnable) and produce the same result more than once (repeatable); it should allow an investigator to reobtain the published results (reproducible) while being easy to use, understand and modify (reusable), and it should act as an available reference for any ambiguity in the algorithmic descriptions of the article (replicable).



# Should Research software be FAIR?

Let us know in the Chat!  
Yes/No/Some not all/Maybe



# The path towards implementation



# Findable

**F1.** Register the software in relevant registry with an assigned DOI

**General** repositories such as Zenodo and Github

- **Language** specific; Python Package Index (PyPI) <https://pypi.org/>
- **Domain** specific; <https://biocontainers.pro/>

# Findable

**F1.** Register the software in relevant registry with an assigned DOI

**General** repositories such as Zenodo and Github

- **Language** specific; Python Package Index (PyPI) <https://pypi.org/>
- **Domain** specific; <https://biocontainers.pro/>

**F2.** Annotate software using domain-agnostic or domain-specific controlled vocabularies

- **The Software Ontology**
- **EDAM-** Ontology of bioscientific data <https://edamontology.org/page>
- **OntoSoft**
- **More @FAIRsharing.org**

# Findable

## **F3.** Include software citation with metadata standards

- **The Citation File Format (CFF)**
- **A CodeMeta instance file**
- **Bi.tools Schema**
- **Bioschemas Tool profile**

# Findable

## F3. Include software citation with metadata standards

- **The Citation File Format (CFF)**
- **A CodeMeta instance file**
- **Biotoools Schema**
- **Bioschemas Tool profile**

CiteAs<sup>[1]</sup>  
alpha

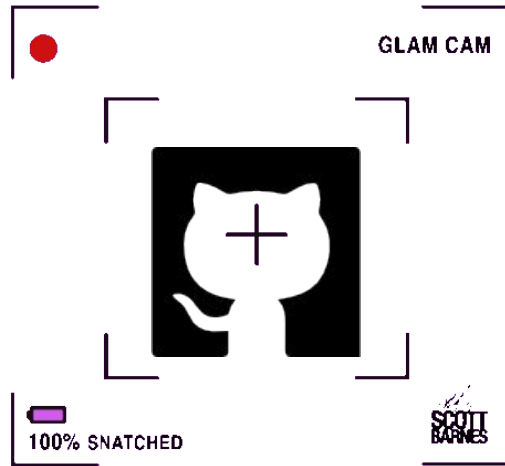
All research products deserve credit.

Get the correct citation for diverse research products, from software and datasets to preprints and articles.

Paste a URL, DOI, arXiv ID, or any search term (e.g. software name/abbreviation)

Examples: <http://yt-project.org> <https://cran.r-project.org/web/packages/stringr> [More examples](#)

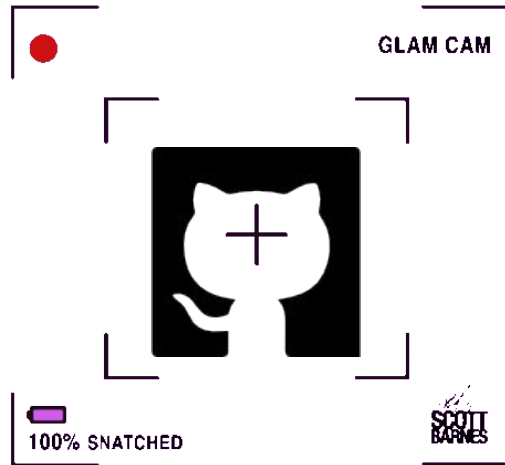
# Accessible



## Take a snapshot from Github

Software stored on Github is accessible for use, reuse and allows for engagement with the community and **versioning**.

# Accessible



## Take a snapshot from Github

Software stored on Github is accessible for use, reuse and allows for engagement with the community and **versioning**.



## Deposit in Zenodo

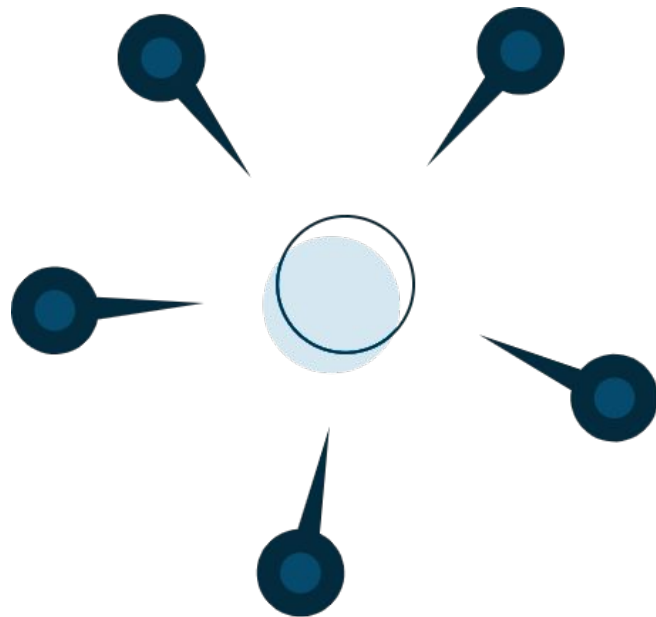
Zenodo offers archival (~20 years), **PID** and opportunity for reproducibility.



# Interoperable

---

- Rich metadata is key!
- The use of Common Workflow Language (CWL), or Workflow Description Language (WDL) enables the interoperability between different pieces of software and workflow platforms
- Containers (e.g. use Docker, singularity) allows for accessibility across different operating systems and environments i.e. software portability.



# Reuse

## License

- License should be as open as possible
- Add clear license, human and machine readable e.g. **Software Package Data Exchange standard**
- License of software components should be compatible



[Get Started](#) [FAQ](#) [Developers](#) [Specification](#) [Resources](#) [Supporters](#) [API](#)

## REUSE SOFTWARE

We make licensing easy for humans and machines alike. We solve a fundamental issue that Free Software licensing has at the very source: what license is a file licensed under, and who owns the copyright? **Adopting our recommendations is as easy as one-two-three!**



**REUSE**  
SOFTWARE

1. Choose and provide licenses
2. Add copyright and licensing information to each file
3. Confirm REUSE compliance

# Reuse

## License

- License should be as open as possible
- Add clear license, human and machine readable e.g. **Software Package Data Exchange standard**
- License of software components should be compatible

## Provenance

- Provenance information with controlled vocabularies e.g. **PROV-O**
- Credit attribution
- How to cite and contribute

# FAIR software summary

1. Deposit in publicly accessible repositories <https://software.ac.uk/choosing-repository-your-software-project>
2. Use a version control system to easily track changes and versions; Github, Gitlab, Bitbucket,
3. Use of containers for software portability; Docker, Singularity
4. Describe with rich metadata including dependencies, with controlled vocabulary: Software Ontology, EDAM
5. Explain the intended use and conditions of functionality of the software
6. Add a license, Apache-2.0 and MIT are permissive licenses with few restrictions, allowing reuse.  
<https://choosealicense.com/> // <https://tldrlegal.com/>
7. Register your code in a community <https://github.com/NLeSC/awesome-research-software-registries>
8. Store snapshots of your software with PIDs <https://guides.github.com/activities/citable-code/>
9. Enable proper citation for your software; CodeMeta and the Citation File Format were specifically designed to enable citation of software
10. **FAIR Software should operate on and deliver FAIR Data!**

# Why should we do it?



# Software is fundamental to research

## It's impossible to conduct research without software, say 7 out of 10 UK researchers

Posted by s.hettrick on 4 December 2014 - 8:07am

By **Simon Hettrick**, Deputy Director.

No one knows how much software is used in research. Look around any lab and you'll see software – both standard and bespoke – being used by all disciplines and seniorities of researchers. Software is clearly fundamental to research, but we can't prove this without evidence. And this lack of evidence is the reason why we ran a survey of researchers at 15 Russell Group universities to find out about their software use and background.



### Tags

- Simon Hettrick
- Policy research
- Research
- Surveys
- Demographics
- Policy

### Headline figures

- 92% of academics use research software
- 69% say that their research would not be practical without it
- 56% develop their own software (worryingly, 21% of those have no training in software development)
- 70% of male researchers develop their own software, and only 30% of female researchers do so

<https://www.software.ac.uk/blog/2014-12-04-its-impossible-conduct-research-without-software-say-7-out-10-uk-researchers>

# Irreproducibility in research

Factors for irreproducible research include:

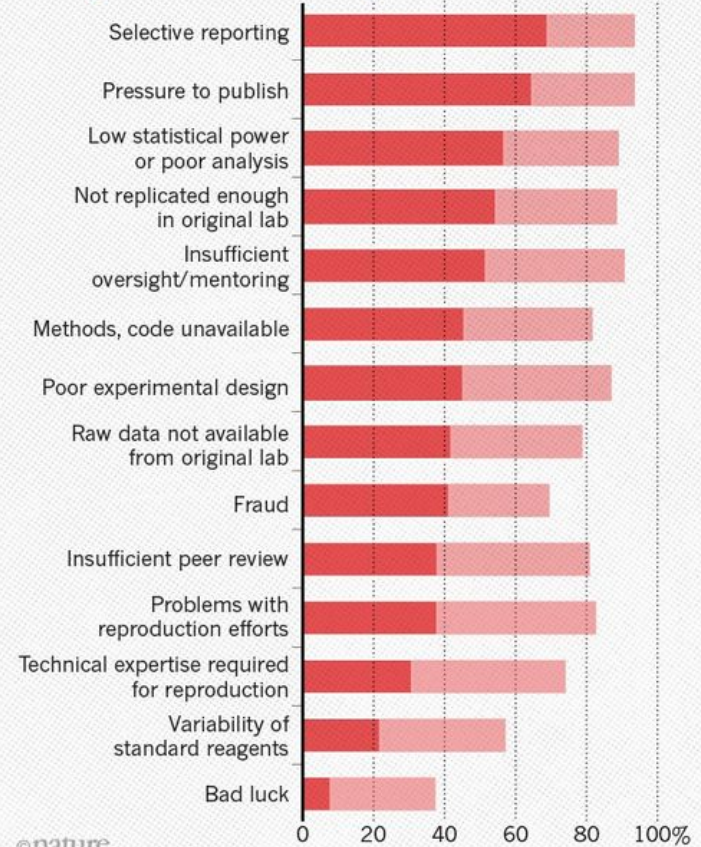
- Selective reporting
- Raw data not available
- Method, code unavailable!

<https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970>

## WHAT FACTORS CONTRIBUTE TO IRREPRODUCIBLE RESEARCH?

Many top-rated factors relate to intense competition and time pressure.

● Always/often contribute ● Sometimes contribute



# Irreproducibility in research



## Experimenting with reproducibility: a case study of robustness in bioinformatics

Yang-Min Kim , Jean-Baptiste Poline, Guillaume Dumas

*GigaScience*, Volume 7, Issue 7, July 2018, giy077, <https://doi.org/10.1093/gigascience/giy077>

**Published:** 28 June 2018 **Article history** ▼

ferent, it is “robustness”. If we used different data but with the same code, it is “replicability”. Last, using different data and different code is referred as “generalizability”. Here, we primarily elaborate on reproducibility and robustness and acknowledge that new datasets or hardware environments introduce additional hurdles [7]. Reproducibility is a key first step. For example, among the 400 algorithms published during the major artificial intelligence conferences, only 6% offered the code [8]. Even when authors provide data and code, the outcome can vary either marginally or fundamentally [9]. Tackling irreproducibility in bioinformatics thus requires considerable effort beyond code and data availability, an effort that is still poorly recognized



**Challenges.....**



# Challenges & Discussion



Let's head to shared notes!  [shiny.link/2h3X2j](https://shiny.link/2h3X2j) 

# Ongoing efforts

---

- FORCE11 Software Citation Working Group ([SCWG](#))
- RDA- FAIR 4 Software WG ([FAIR4S](#))
- Research Software Alliance ([RESA](#))
- Software Sustainability Institute ([SSI](#))

📌 If you want to stay up-to-date on FAIR implementation solutions:

**Register:**

<https://www.scilifelab.se/event/fairpoints/>



# Resources

- Special Issue: Emerging FAIR Practices. Issue Editors: Barend Mons, Erik Schultes & Annika Jacobsen <https://direct.mit.edu/dint/issue/2/1-2>
- Software vs. data in the context of citation: DOI [10.7287/peerj.preprints.2630v1](https://doi.org/10.7287/peerj.preprints.2630v1)
- Software citation principles DOI <https://doi.org/10.7717/peerj-cs.86>
- Sharing interoperable workflow provenance: A review of best practices and their practical application in CWLProv DOI <https://doi.org/10.1093/gigascience/giz095>
- Five recommendations for FAIR Software <https://fair-software.eu/>
- FAIR Principles for Research Software (FAIR4RS Principles) <https://rd-alliance.org/group/fair-research-software-fair4rs-wg/outcomes/fair-principles-research-software-fair4rs>
- Taking a fresh look at FAIR for research software DOI <https://doi.org/10.1016/j.patter.2021.100222>
- The role of metadata in reproducible computational research DOI <https://doi.org/10.1016/j.patter.2021.100322>
- RDA Webinar: FAIR Principles for Research Software (FAIR4RS WG) DOI <https://doi.org/10.5281/zenodo.5524726>
- Toward Better Research Software DOI <https://doi.org/10.5281/zenodo.4551441>
- FAIR4RS WG subgroup community consultation March 2021 DOI <https://doi.org/10.5281/zenodo.4635410>
- From FAIR research data toward FAIR and open research software DOI <https://doi.org/10.1515/itit-2019-0040>
- FAIR for Research Software (FAIR4RS) publication list on Zenodo <https://zenodo.org/communities/fair4rs?page=1&size=20>

# The End

