Four recommendations to make research code more visible, citable and FAIR

PRESENTED BY

Paula Andrea Martinez
Software Project Coordinator
Australian Research Data Commons
0000-0002-8990-1985

16 June 2022

DOI: 10.5281/zenodo.6640422

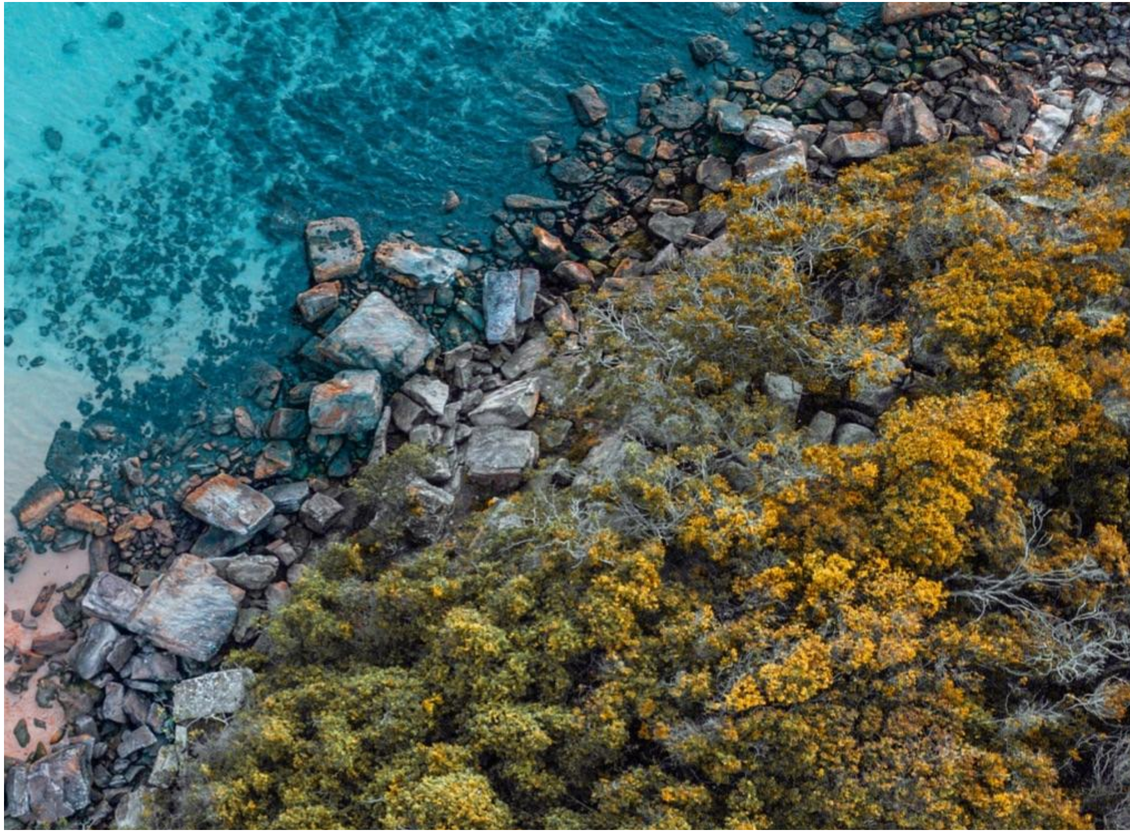
ARDC
Australian Research Data Commons

NCRIS
National Research Infrastructure for Australia
An Australian Government Initiative

ARDC is enabled by NCRIS

ACKNOWLEDGEMENT OF COUNTRY

We acknowledge and celebrate the First Australians on whose traditional lands we meet, and we pay our respect to their elders past, present and emerging.



ARDC

Australian Research Data Commons

Purpose

To provide Australian researchers with competitive advantage through data.

Mission

To accelerate research and innovation by driving excellence in the creation, analysis and retention of high-quality data assets.

NCRIS
National Research Infrastructure for Australia
An Australian Government Initiative

ARDC is enabled by NCRIS

> “Research is now fundamentally connected to software”
>
> “It permeates every aspect of the conduct of research.”

Ten computer codes that transformed science Perkel, 2021

10.1038/d41586-021-00075-2

Neil Chue Hong, Director

SSI, Software Sustainability Institute

“Let's acknowledge that
we need software
- to search for relevant data sources
- to extract relevant data
- to assess the quality of the data
- to analyse and combine data
- to summarise the evidence”

Martinez, 2022
https://twitter.com/orchid00/status/1493080140338778112

GOALS

- Practical suggestions that contribute to making research software and its source code more discoverable, reusable and transparent.
- Discuss the alignment of these recommendations with the FAIR Principles for research software (FAIR4RS).

NOTES

- We are not covering software development best practices.
- The recommendations are technology independent.

WHAT IS RESEARCH SOFTWARE?

RESEARCH SOFTWARE

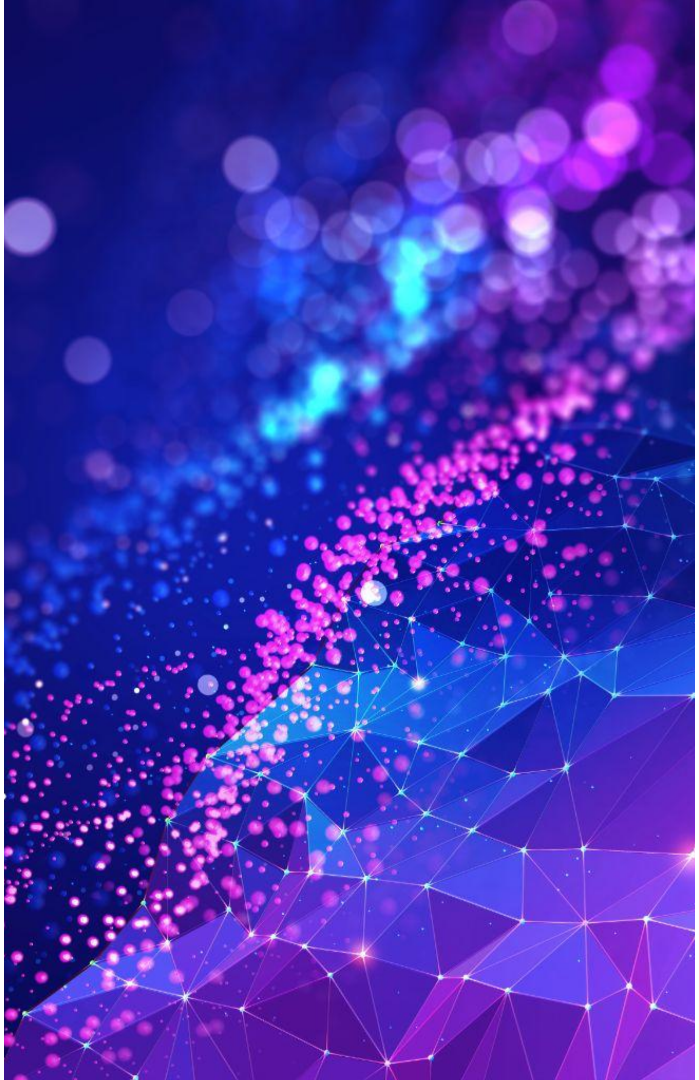
Source code files, algorithms, scripts, computational workflows and executables that were created in either of two categories:

A. Within a research project as a by-product to do the research, or

B. Through intentional development of a software product for general use in research by one or more projects.

(adapted from Gruenpeter et al. 2021).

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



ARDC RESEARCH SOFTWARE PROGRAM

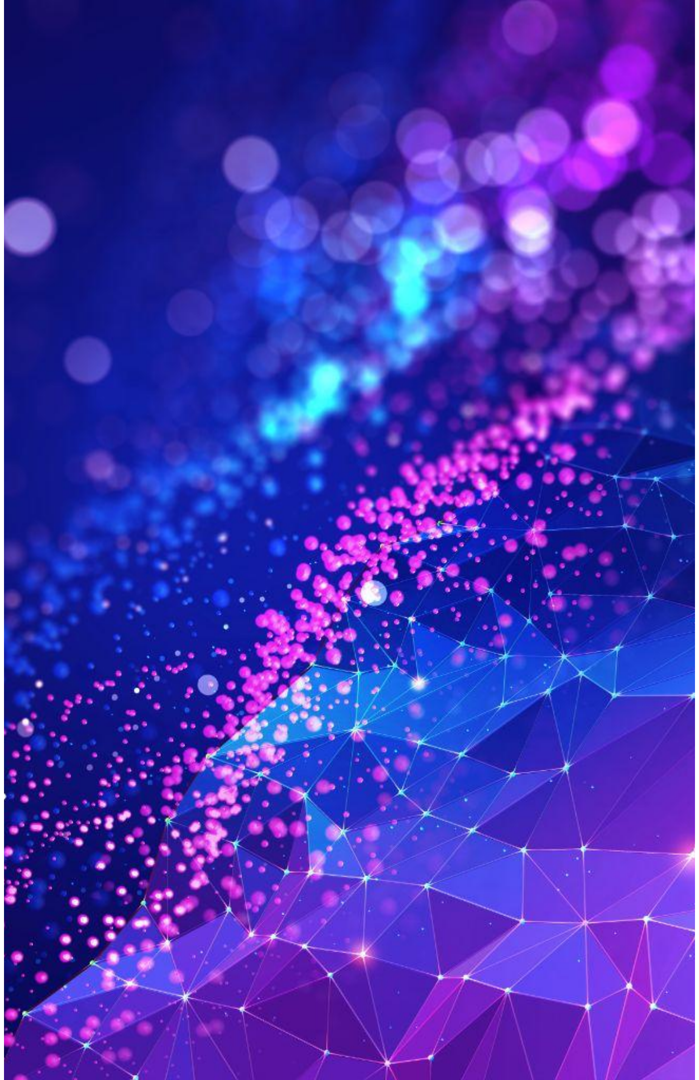
Making research software a first-class research output ensures Australia can maximise the value that software represents.

ARDC NATIONAL RESEARCH SOFTWARE AGENDA

Has three layers of distinct action to **SEE**, **SHAPE** and **SUSTAIN** research software

SEE: addresses the need for all forms of research software, particularly analysis code, to become more visible — shared, published, cited and acknowledged.

Target beneficiaries are researchers
Impact is research integrity



FAIR RESEARCH SOFTWARE #FAIR4RS

FINDABLE

ACCESSIBLE

INTEROPERABLE

REUSABLE

https://www.rd-alliance.org/groups/fair-research-software-fair4rs-wg

https://doi.org/10.15497/RDA00068

SOFTWARE IS NOT JUST DATA

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

Warrant changes to several of the data-oriented principles.

Lamprecht et al, 2020 https://doi.org/10.3233/DS-190026.

FAIR PRINCIPLES FOR RESEARCH SOFTWARE

F: Software, and its associated metadata, is easy for both humans and machines to find.

A: Software, and its metadata, is retrievable via standardized protocols.

I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.

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

Chue Hong et al 2022, https://doi.org/10.15497/RDA00068

ABOUT THE FOUR RECOMMENDATIONS

Jiménez RC, Kuzak M, Alhamdoosh M *et al.* Four simple recommendations to encourage best practices in research software [version 1; peer review: 3 approved]. *F1000Research* 2017, **6**:876 (https://doi.org/10.12688/f1000research.11407.1)

https://softdev4research.github.io/4OSS-lesson/

DOI https://doi.org/10.12688/f1000research.11407.1

1. Develop source code in a version controlled repository from the beginning of the project

- A software repository or "repo" is a storage location for software
- Version Control System - track changes
- Build software “together”
- Private vs Public
- Self-hosted vs web-based, platform independent
- Saves you time before publication
- Increases **transparency** through community scrutiny

DOI https://doi.org/10.12688/f1000research.11407.1
https://alternativeto.net/category/developer-tools/version-control-system

1. **Develop source code in a version controlled repository from the beginning of the project**

alternativeto.net/category/developer-tools/version-control-system

readme.so

1. **Develop source code in a version controlled repository from the beginning of the project**

   ACCESSIBLE AND REUSABLE

   A1.1 Software is retrievable using a protocol open, free, and universally implementable.
   R1.2. Software is associated with detailed provenance.

2. Adopt a licence and comply with the licence of third-party dependencies

- Terms of use, change, contribute, redistribute
- Talk to the Copyright Officer
- choosealicense.com
- Depends on your needs, community standards
- The SPDX license list spdx.org/licenses
- Builds trust and transparency in how software is created, distributed, and consumed
- ARDC Research Software Rights Management Guide https://doi.org/10.5281/zenodo.5003962

DOI https://doi.org/10.12688/f1000research.11407.1

> “Always license your code. Unlicensed code is closed code.”

Jake VanderPlas, Director of Research in the Physical Sciences, the University of Washington’s eScience Institute. The Whys and Hows of Licensing Scientific Code. https://www.astrobetter.com/blog/2014/03/10/the-whys-and-hows-of-licensing-scientific-code

https://choosealicense.com/no-permission

2. Adopt a licence and comply with the licence of third-party dependencies

INTEROPERABLE and REUSABLE

I2. Software includes qualified references to other objects

…can be understood, modified, built upon, or incorporated into other software.

R1.1. Software is given a clear and accessible license.

R3. Software meets domain-relevant community standards.

DOI https://doi.org/10.12688/f1000research.11407.1

“Generally when scientists make their code public, they do so because they want it to be free to use and as useful as possible for as many people as possible.”

Jake VanderPlas, Director of Research in the Physical Sciences, the University of Washington’s eScience Institute. The Whys and Hows of Licensing Scientific Code. https://www.astrobetter.com/blog/2014/03/10/the-whys-and-hows-of-licensing-scientific-code/

3. Define clear and transparent contribution, governance and communication processes

- Communication channels and ways to get involved
- End users
- Your collaborators
- Evaluators/ Reviewers
- https://galaxyproject.org/community
- https://softdev4research.github.io/4OSS-lesson/04-contributions/index.html

DOI https://doi.org/10.12688/f1000research.11407.1

> “Generally when scientists make their code public…
> They want others to not only use it,
> but also extend it,
> fix bugs,
> incorporate it into their own research code,
> and thereby make it even more useful to more people.”

Jake VanderPlas, Director of Research in the Physical Sciences, the University of Washington’s eScience Institute.
The Whys and Hows of Licensing Scientific Code.
https://www.astrobetter.com/blog/2014/03/10/the-whys-and-hows-of-licensing-scientific-code

3. Define clear and transparent contribution, governance and communication processes

INTEROPERABLE and REUSABLE

- I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards.
- R3. Software meets domain-relevant community standards.

DOI https://doi.org/10.12688/f1000research.11407.1

4. Make software easy to discover by providing software metadata via a community registry

- A Registry stores information about the software
- Facilitates discoverability
- Increases the visibility of the project
- Increases the chances of collaboration, reuse, and improvement

DOI https://doi.org/10.12688/f1000research.11407.1

4. Make software easy to discover by providing software metadata via a community registry

- Metadata might include information like the source code location, contributors, licence, version, identifier, references and how to cite the software. https://codemeta.github.io
- Exposes metadata in a machine readable format
- https://github.com/NLeSC/awesome-research-software-registries
- Domain specific first
- https://www.software.ac.uk/which-journals-should-i-publish-my-software
- https://citation-file-format.github.io a list of metadata fields

DOI https://doi.org/10.12688/f1000research.11407.1

4. Make software easy to discover by providing software metadata via a community registry

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

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.

DOI https://doi.org/10.12688/f1000research.11407.1

4. Make software easy to discover by providing software metadata via a community registry

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

DOI https://doi.org/10.12688/f1000research.11407.1

FOUR RECOMMENDATIONS TO MAKE RESEARCH CODE VISIBLE

1. Develop source code in a version controlled repository from the beginning of the project
2. Adopt a licence and comply with the licence of third-party dependencies
3. Define clear and transparent contribution, governance and communication processes
4. Make software easy to discover by providing software metadata via a community registry

DOI https://doi.org/10.12688/f1000research.11407.1

[Recommends] "4. Take steps to make research data and other research-relevant digital objects from public funding understandable and re-usable in the long term, including through the provision of high quality human-readable, machine-actionable, and open metadata and adequately maintained and supported bespoke algorithms, code, software, and workflows essential for re-use of data as free and open source."

2021. OECD Recommendation of the Council concerning Access to Research Data from Public Funding *https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0347*

BENEFITS FOR AUTHORS

- Increase discoverability and **visibility** of research software
- **Engages** authors and user **communities**
- Provides **recognition** for contributions
- Builds **trust** among users

DOI https://doi.org/10.12688/f1000research.11407.1

HOW CAN YOU ENABLE THESE RECOMMENDATIONS?

1. Endorsement: agreeing to support the recommendations without a formal process for implementation.
2. Promotion: actively publicising and incentivising the recommendations within and beyond the organisation.
3. Compliance to formally implement the recommendations within the organisation, with ongoing monitoring and public reporting if possible.

DOI https://doi.org/10.12688/f1000research.11407.1

PRESENTED BY

Dr Paula Andrea Martinez

Software Project Coordinator,
Software *Programme, ARDC*

Slides DOI: [10.5281/zenodo.6640422](https://doi.org/10.5281/zenodo.6640422)

paula.martinez@ardc.edu.au

@orchid00

linkedin.com/in/pambio/

[0000-0002-8990-1985](https://orcid.org/0000-0002-8990-1985)

Subscribe to the **ARDC CONNECT** newsletter

THANK YOU

ardc.edu.au

contact@ardc.edu.au

+61 3 9902 0585

@ARDC_AU

Australian-Research-Data-Commons

ARDC
Australian Research Data Commons

NCRIS
National Research Infrastructure for Australia
An Australian Government Initiative

ARDC is enabled by NCRIS