# Report: Data is Only Half the Battle

## Birds of a Feather Session at eResearch Australasia 2021

31/10/2021

DOI: 10.5281/zenodo.5636530

## How to cite this report

Martinez, Paula, Crosby, Vanessa, Honeyman, Tom, Guillou, Stéphane & Moresi, Louis. (2021). Report: Data is Only Half the Battle. Zenodo. https://doi.org/10.5281/zenodo.5636530

## Report Editors

| | | | |
|---|---|---|---|
| Dr Paula Andrea Martinez | Australian Research Data Commons | Software Project Coordinator | 0000-0002-8990-1985 |
| Dr Vanessa Crosby | University of New South Wales, Sydney | Associate Director, Scholarly Communications and Repositories | 0000-0002-9474-6553 |
| Dr Tom Honeyman | Australian Research Data Commons | Software Program Manager | 0000-0001-9448-4023 |
| Stéphane Guillou | The University of Queensland | Technology Trainer | 0000-0001-8992-0951 |
| Louis Moresi | The Australian National University | Professor | 0000-0003-3685-174X |

## Workshop contributors and their institutions

The organisers want to acknowledge the attendees and contributors to this session. Listed below and ordered alphabetically by first name.

| **Name** | **Institution** |
|---|---|
| Adrian Chew | UNSW |
| Alex Reid | AeRO/UWA |
| Amanda Flanders | CQUniversity |
| Andrew Harrison | Monash University |
| Andrew Treloar | ARDC |
| Anitha Kannan | ARDC |
| Avril Clarkson | UNSW |
| Briget Lander | Plant & Food Research NZ |
| Brock Askey | Bond University |
| Carmel Walsh | ARDC |
| Catherine Randalls | JCU |
| Cesar Herrera | Griffith University |

| | |
|---|---|
| Chris Seal | University of Auckland |
| Claire Trenham | CSIRO |
| Danny Kingsley | Flinders University |
| Dave Saunders | DST |
| Duncan Smith | UNSW |
| Edmond Chuc | TERN/UQ |
| Emma Flukes | IMAS (UTAS) |
| Erin Kenna | CSIRO |
| Fei Yu | The University of Queensland |
| Francis Gacenga | University of Southern Queensland |
| Gareth Williams | CSIRO |
| Gayle Crooks | The University of Queensland |
| Gene Melzack | University of Melbourne |
| Harko Werkman | UTAS |
| Ian Duncan | ARDC |
| Jake Surman | UNSW |
| James Wilmot | UNSW |
| Jenny Rowland | Geoscience Australia |
| Jianzhou Zhao | Intersect Australia |
| Jo Savill | ARDC |
| Johan Gustafsson | University of Melbourne |
| Julia Martin | ARDC |
| Karin Quadros | Monash University |
| Kayla Maloney | University of Sydney |
| Keith Russell | ARDC |
| Kerry Levett | ARDC |
| Kristal Spreadborough | University of Melbourne |
| Kyle Hemming | Intersect Australia (UC) |
| Len Smith | ANU |
| Liz Stokes | ARDC |
| Luc Betbeder-Matibet | UNSW |
| Lyle Winton | University of Melbourne |
| Mike Lynch | UTS |
| Myra Cheng | ARDC |
| Pascal Elahi | Pawsey |
| Pascal Tampubolon | UTS |
| Patricia Hayman | CQ University |
| Paula Andrea Martinez | ARDC |
| Peter Sefton | The University of Queensland |
| Philippa Frame | QUT |
| Rebecca Farrington | University of Melbourne |
| Rory Chen | UNSW |

| | |
|---|---|
| Rosemarie Sadsad | University of Sydney |
| Sandra Silcot | Indigenous Data Network |
| Siobhann McCafferty | ARDC |
| Stefanie Kethers | ARDC |
| Stéphane Guillou | The University of Queensland |
| Steve Quenette | Monash University |
| Susan Wilson | The University of Queensland |
| Tim Sherratt | GLAM Workbench / UC |
| Tom Honeyman | ARDC |
| Vaidehi Hardikar | UTAS |
| Vanessa Crosby | UNSW |
| Varvara Efremova | UNSW |
| Wayne spagnol | JCU |
| Yvette Wharton | University of Auckland |

# Table of Contents

# Executive summary

This report describes the Birds of a Feather (BOF) session at the eResearch Australasia Conference 2021, titled *Data is Only Half the Battle,* hosted on October 14, 2021. The first objective of the BOF was to collect and communicate relevant guidance, policy and other materials around digital objects different from data, particularly for research software. The second aim was to identify champions to form a Task Force to socialise existing resources and guidance via Australian Research institutions.

First, the report records the pre-work coordinated by the organisers leading to the event. Then, it summarises the discussions and shared resources to support research software authors.

To get involved or for any questions about the BOF or this report email contact@ardc.edu.au with the subject: “Research Software Authors Support”.

# 1. Introduction

This report describes the Birds of a Feather (BOF) session at the eResearch Australasia Conference 2021, titled *Data is Only Half the Battle*, hosted on October 14, 2021. The first objective of the BOF was to collect and communicate relevant guidance, policy and other materials around digital objects different from data, particularly for research software. The second aim was to identify champions to form a Task Force to socialise existing resources and guidance via Australian Research institutions.

First, the report records the pre-work coordinated by the organisers leading to the event. Then, it summarises the discussions and shared resources to support research software authors. These resources were initially collected by the organisers and subsequently attendees of the workshop and editors collaborated with further resources. For information and dissemination to those who attended the session and those interested in the topic, but unable to attend, the editors summarised the discussion and shared resources in this report.

## 1.1. Abstract

**Data is Only Half the Battle**

Type: Birds of a Feather sessions 60 minutes
Category: Future focussed eResearch Strategy

Research reproducibility is a cornerstone of the scientific process. Although describing, preserving, and sharing data has become increasingly common, the research software and workflows used to extract knowledge from that data aren't usually well curated, preserved or cited. Research data management without the accompanying processing of research software management is only half the battle. The value of research software and code can be leveraged by making it reusable for others to improve, with clear documentation and metadata that can be used for citation, curation and maintenance. Ensuring that research software remains accessible is paramount to enable better research.

Do you provide support for digital objects: curation, citation, provenance, software management plans, ORCIDs, identifiers, metadata, digital preservation services, computational reproducibility, best practices, interoperable systems and sustainable outcomes? If you are the contact person for these topics, we are looking for you!

We would like to share with you national and international resources. That will help communicate relevant guidance, policy and other materials around digital objects different from data. Come along and find out about emerging software citation practices, publication, preservation and licences.

The aim of this BOF is to identify champions to form a Task Force to socialise existing resources and guidance.

https://conference.eresearch.edu.au/events/data-is-only-half-the-battle

## 1.2. Coordination

The eResearch Australasia Conference 2021 covers topics across technical, research activities and strategic initiatives planned for the eResearch sector in Australasia. The program committee invites submissions from the sector on an annual basis.

The success of a workshop is defined by the activities that help shape it. In early July 2021, ARDC's Paula Martinez, Tom Honeyman and Liz Stokes started discussing the proposal for a "Birds of a Feather" (BOF) session. In mid-July we contacted institutional representatives to learn about institutional guidance for promoting software citation, registries, provenance, preservation, and best practices, including the FAIR for Research Software principles (FAIR4RS). The response indicated that there were gaps in guidance for the sector. In late July, we invited institutional representatives to be involved in shaping the proposal and to contribute the resources already developed at their institutions. In early August, a team of seven organisers (see About the organisers) collaborated on the writing and reviewing the BOF proposal.

The BOF was accepted in September, and subsequently two 1-hour sessions were scheduled to discuss the presentation content, mode and times, and sent targeted invitations requesting contributions to the discussion. The organisers used emails, Twitter and Slack channels to invite people to attend the BOF. Content was prepared using a slide deck, also a form was provided to collect community resources and help with their categorisation. A week before the session, the organisers had the presentation content ready and a fixed agenda was available to share during the BOF.

During the BOF session, two levels of participation were used. First, the organisers prepared to convey information for attendees to consume independently of one another. Then, the presenters provided scaffolding for the discussion and invited attendees to contribute the resources known to them in specific areas of interest (see section Discussion). The guides referenced during the discussion were added to the collection of resources (see section Shared resources). This report was written to amplify the efforts after the BOF, the organisers invited all the participants and interested people to co-author this report (see section Report Editors).

## 1.3. About the organisers

This BOF was skillfully coordinated by a team of 7 co-leads from four research institutions.

Table 1. BOF organisers details

| Name | Role | Affiliation | ORCID |
|---|---|---|---|
| Paula Andrea Martinez | Software Project Coordinator | Australian Research Data Commons | https://orcid.org/0000-0002-8990-1985 |
| Gene Melzack | Data Curator with Scholarly Services | The University of Melbourne | https://orcid.org/0000-0002-6321-9953 |
| Dr Vanessa Crosby | Associate Director Scholarly Communications and Repositories | The University of New South Wales | https://orcid.org/0000-0002-9474-6553 |
| Luc Betbeder-Matibet | Director, Research Technology Services | The University of New South Wales | https://orcid.org/0000-0003-4065-5784 |
| Stéphane Guillou | Technology Trainer | The University of Queensland | https://orcid.org/0000-0001-8992-0951 |
| Tom Honeyman | Software Program Manager | Australian Research Data Commons | https://orcid.org/0000-0001-9448-4023 |
| Liz Stokes | Senior Research Data Skills Specialist | Australian Research Data Commons | https://orcid.org/0000-0002-2973-5647 |

## 1.4. About the attendees

The BOF was targeted at people who provide support for digital research objects, e.g. curation, citation, provenance, software management plans, ORCIDs, identifiers, metadata, digital preservation services, computational reproducibility, best practices, interoperable systems, FAIR and sustainable outcomes. During the planning, the organisers envisioned attendees to be either people who are part of an eResearch team, work on a Research Software Engineering team, are data or digital librarians, are researchers with an interest in these practices and infrastructures.

The BOF was attended online by 67 people who registered their attendance during the session: 58% of attendees were affiliated with Australian Universities, the other 42% were a mix of discipline-specific project members and research infrastructure providers (e.g. ARDC, CSIRO, Intersect, TERN, PAWSEY and DST).

The BOF successfully drew the contributions of a wide range of community members. Of the total group of attendees, 13% were affiliated with institutional libraries. Analysing the role titles, 32% had the words Director, Manager or Senior in their title, 10% had the word Software in their title, and 67% had the word research in their role title. This demonstrates the diverse background and wide range of interests across the attendee group.

## 1.5. Feedback

Table 2. Feedback from organisers

| | |
|---|---|
| Liz Stokes | “Collaborating in the Data is only Half the Battle BoF was genuinely exciting, and with a strong engagement with research software authors. Tremendous and thorough coordination. It was a pleasure to work with everyone on and in this BoF session - certainly one of the highlights of this year's conference” |
| Stéphane Guillou | “Well done everyone! Very impressed with the level of engagement during the session. We could have gone on for another hour” |
| Tom Honeyman | “Congratulations, that was an absolutely fantastic session” |

Table 3. Feedback from attendees

| | |
|---|---|
| Kerry Levett | “This was an excellent session, and a masterclass in how to engage participants online.” |
| Anonymous | “So as a software developer myself I found the information in your slides very interesting and I can see why you had such an engaged session” |
| Rory Chen | “A wonderful session!” |
| Siobhann McCafferty | “Really interesting. Thanks all!” |
| Pascal Tampubolon | “Great session, thank you!” |
| Cesar Herrera Acosta | “Thanks so much for organizing and putting all this information together” |
| Varvara Efremova | “Fantastic session, thanks!” |
| Michael Lynch | “Thanks everyone, great session” |
| Andrew Treloar | “my congratulations for an excellent session. The level of energy and enthusiasm in the (virtual) room was palpable. Well done for pulling it together and dealing with last minute absences from presenters!” |
| Plus many other thank you messages | |

# 2. Summary of Discussion

## 2.1. Topics

The discussion focused on the current barriers researchers face when sharing, curating, preserving and citing research software, as well as the skills, infrastructure and support necessary to build a FAIR research software culture. This section describes assertions of need in respect to eight topics: incentives, software citation, platforms and infrastructure, software consultancy services, researchers who code, code sharing with reuse in mind, reproducibility and portability, maintenance and preservation.

### 2.1.1. Incentives

Across the board, participants emphasised the need for institutions to recognise and reward the production of research software and their authors. Anecdotally, citations are higher for papers with associated software in some fields. Van Noorden, Maher & Nuzzo, 2014 stated "the vast majority of the most-cited research of all time describe experimental methods or software that have become essential in their fields". A correlation between software availability and citation metrics could be highlighted to researchers as an incentive to publish their software.

The lack of recognition for research software in hiring and promotions processes was highlighted as a barrier. Adoption of UKRI-style policies (Software for research communities, guidance for applicants to funding opportunities from UKRI) for researchers with a track record in coding would be a welcome development in Australia. Awards for software contributing to research are a way of incentives, for example the French Open Science Free Software Award and the Digital Preservation Award from the Software Sustainability Institute. There is some evidence of shifting practice in this area in Australia with Monash Business School now formally considering research software as an output for promotion purposes. Also, The Statistical Society of Australia Victorian Branch (SSA Vic) with the Di Cook Award, an open-source statistical software award for students.

The lack of software metrics and indicators of prestige comparable to journal articles is a barrier to more widespread recognition of research software in promotions. Development of robust metrics is reliant on findable, accessible, interoperable and reusable (FAIR)-compliant metadata for published software to support software citation, and on enterprise systems including repositories and Current Research Information Systems (CRIS) capturing metadata for research software to enable tracking and reporting. Existing software metadata as examples in this area are the bioschemas Draft for Computational Tool for describing a Software Application in the Life Sciences, a more generic example is the CODEMETA file which describes the metadata associated with a software object, and the DESCRIPTION file which contains basic information about an R package.

While measuring software citation (see section Software Citation) and reuse is essential to incentivising the creation of research software, the community is keen to avoid the perverse incentives and monetisation of software citation metrics by proprietary platforms that currently plague traditional publications. For example, rewarding volume may result in unnecessarily modular production of code, similar to the phenomenon of "salami slicing" seen in publication. Any efforts by the community to incentivise and reward software production should champion open citation standards and responsible use of metrics from the outset to avoid these outcomes.

### 2.1.2. Software Citation

Participants noted the importance of embedding a culture of software citation in order to enable reward and recognition of research software development and their authors. Currently, software publication and citation practices vary considerably between disciplines. Some disciplines, including biosciences and geosciences, link open source software packages to articles that outline the method associated with the software. This enables persistent identifiers, such as DOIs to be minted for software and assists in tracking citations. At the same time, other participants noted the problems associated with only "counting" journal articles describing software in incentives schemes and not rewarding software as research outputs in their own right.

Barriers to software citation include lack of awareness of software registration and assignment of persistent identifiers as a common practice to aid findability. In addition to poor awareness of software licences, to enhance reusability. A long standing issue with sharing code without a licence is that by default others have no permission to use, modify, or share the software. A licence defines what others can do with the software. To enhance software citation, embedded mechanisms for authors are needed, for example, R's inbuilt citation mechanism, which allows developers to add citation information to their R package provides a useful model of good practice. The Citation File Format (CFF) is now more broadly adopted thanks to integration with various systems that are currently in use by people who write code; including software version control platform GitHub. To learn more about Software Citation please visit FORCE11 Software Citation Principles and Software Citation resources below.

### 2.1.3. Platforms and Infrastructure

Researchers are recommended to deposit their data and software in a repository that specialises in their scientific domain. Global and domain specific registries offer tailored mechanisms to share and host data and software, maximising the probability that the digital object will be FAIR. For example, in Bioinformatics there are current platforms or mechanisms that are in use, including Bioconda (a package manager specialising in bioinformatics software), biocontainers (a registry of bioinformatics containers and workflows including metadata and statistics), use shared cloud computing (e.g. institutional, AWS, Galaxy AU) etc. Integrating FAIR registries for tools, containers and workflows with the community platforms that are capable of deploying them would provide much needed structure in this space, while supporting transparency, preservation, reuse and citation.

If a domain specific repository is not available the next recommendation is to use an institutional repository, with rich metadata to enable FAIR. Some institutions (see section Code development infrastructure at institutions in Australia) now offer institutional-wide services for internal and external software sharing, which is different from publishing. An example of a publishing infrastructure for software is CSIRO Data Access Portal (DAP) that supports publication and search of software as well as data. Another example is AARNet, the Australian network provider for education and research institutions, provides cloud storage to store research data and software, and a cloud environment to write and run code-based processes (Cloudstor and SWAN). These services enable version control, metadata, packaging and distribution. However, it does not integrate DOI minting or Git versioning of the code.

The next recommendation is to use a generic repository when the other options are not available. Generic repositories are Zenodo, Dryad, Mendeley Data, Figshare, etc. Those will require minimum metadata, generate a persistent identifier, a recommended citation and will prompt the owner to use a licence. These generic repositories are usually linked with free and/or self-hosting accounts for version control and software management systems, such as GitHub, GitLab, Bitbucket, GitBucket, Gitea, Gogs, etc. Researchers are often not aware that the problem with generic code and data repositories is that discipline based requirements/standards are not validated by generic repositories, for example, the use of formats and vocabularies that enable findability, accessibility, interoperability and reusability. In addition, generic repositories could operate at odds with institutional IP, security and preservation policies.

The distinction between development systems and registries (that enable i.e. persistent identifiers) above is an "ecosystem of platforms" and they should integrate appropriately; and the pathways to access them should be clearer. A key focus of the community should be the collective alignment to platforms that together form this ecosystem connecting software development and version control with software registration and assignment of persistent identifiers (see section Publishing software / code availability).

### 2.1.4. Software Consultancy Services

Research institutions’ software consultancy services would be a valuable addition to data consultancy services. Currently, advice tends to be provided on an informal basis by research and software support staff. There would be advantages to providing software consultancy as part of established data consultancy services depending on the needs of specific institutions. Formalised training would be required to develop these services and ensure that staff had the required skills sets and knowledge.

Topics covered by a software consultancy could include:

- Best practice in making code open source, including code review
- Managing sensitive data when making code open source
- Considerations for licensing and commercialisation of software
- Assistance in planning for codebase maintenance and overall project sustainability

### 2.1.5. Researchers who code

Many researchers/academics who code would not consider themselves developers, coders and even less so research software engineers because they have not been formally trained as such or do not have enough practical experience. Despite this, researchers who write code to produce research results should be able to warrant access to the code that the research outcome is built upon (Barnes, 2010), for research integrity. This is already a requirement in some journals, such as Nature.

Expectations about the quality and longevity of the code should follow domain-specific agreed standards. The consensus message is that code that supports research results should not be stored exclusively on the researcher's personal workspace, as it would prevent desirable outcomes that include: recognition of that work and effort, improvement through peer review, contribution to building the open-source movement, public engagement with the work and creative repurposing of the work by others.

### 2.1.6. Code sharing with reuse in mind

Where is code available and how to improve its findability and reduce duplication efforts are general topics of interest for software authors and software users (see section Platforms and Infrastructure). Also converging on platforms / standards that provide guidance for documentation, metadata for decision making, specifically to help identify what is mature, maintained and relevant. These indicators could include: programming language; maintenance frequency (last commit, last release); number of contributors; licence; age of project; declared institutional support; etc. See section Apply FAIR to Research Software in particular the FAIR principles for research software.

Code sharing with reuse in mind is intended to increase transparency of research outputs not as an afterthought, but from the inception.

### 2.1.7. Reproducibility and portability

Making a process reproducible on various systems and at various time points can be challenging, in particular because of the fluid landscape of software libraries (also known as “packages”, “extensions” or “modules”) that the process depends on.

On top of code-authoring best practices, the use of computational reproducible environments (with fixed library versions) can help increase the reproducibility of a process, for a specific research project. Python and R both have tools to define such environments and provide others the instructions to recreate an exact copy on a different system. For example, online platforms like Binder provide options to run code in a fixed environment independently of the user’s system.

Containerisation of research software is yet another option to increase reproducibility and portability, packaging not only specific versions of libraries, but also the underlying operating system. Institutional High performance Computing (HPC) providers use shared resources and hence make use of Singularity, a container system to encompass all the requirements of a particular process or software application.

However, the discussion highlighted that it is challenging for researchers to guarantee a high level of reproducibility on top of other responsibilities, especially when it is not part of their existing skillset. A solution could be to involve software engineers and/or eResearch support staff who can focus on that aspect, and plan this kind of resourcing at the project proposal phase. Such written records of requests for specialised staff could then potentially inform hiring decisions at the institution level.

### 2.1.8. Maintenance and Preservation

Challenges associated with long term maintenance and preservation of research software are exacerbated by the mismatch between research software development lifecycles and research project lifecycles. Fixed project-based funding limits the support available for necessary ongoing maintenance of research software. Institutions will need to recognise that code development frequently starts before a research project commences and can be required to continue after project funding ends. This would entail establishing processes for determining responsibility for code maintenance when staff move on, determining whether the software project should continue and, if suitable, ensuring support is in place to maintain code as part of business-as-usual activities.

Ongoing maintenance of software can become an uncompensated workload for researchers/developers. Popular software packages naturally attract higher maintenance loads due to higher volumes of feature requests and bug reports, which can be especially challenging to manage. Guidelines on software maintainability, managing change and managing technical debt would help minimise the burden for researchers. Research software should also have a retirement process where appropriate: this is particularly urgent for prototype tools. Policies and procedures could be put in place to determine which packages are candidates for long term maintenance and preservation and which can be retired. These policies need to be clear to both the developers and code users.

Post-project maintenance funding is essential and needs to be tackled nationally, rather than solely at an institution level. Funding bodies have begun to recognise the problem. For example, the Chan Zuckerberg Initiative has committed to providing more funding for sustaining software (CZI, 2020). More advocacy in this space is needed to encourage other funding bodies to adopt similar policies.

## 2.2. Takeaways

“One important reflection was recognising that a key challenge of making reproducible research software and workflows is a community effort”. - Liz Stokes

“The energy and enthusiasm of participants and the breadth and depth of discussion between them are strong indicators that now is the time to start seriously tackling research software. The knowledge and people are out there, we just need to start cohering in a community.” - Tom Honeyman

# 3. Shared resources

Participants in the BoF were invited to share relevant resources produced by their institutions, or discovered elsewhere. These were grouped into categories by the BoF co-leads. Some items are relevant across multiple categories, in which case they are repeated.

## Software Citation

- ARDC landing page: software citation
- ARDC guide: How to make your software citable
- ARDC presentation: Software publishing, licensing, and citation
- FORCE11 Software Citation Principles
- Article: "Recognizing the value of software: a software citation guide" (arising from the FORCE11 software citation WG)
- Blog post: "#GenR Software Citation Round-up"
- Metadata Standard: Citation File Format (CFF)
    - now supported by GitHub, Zenodo and Zotero (see Smith, 2021 and Fenner, 2021)
- GitHub Guide (zenodo integration): "Making your code citable"
- CiteAs: Citation formating tool
- CrossCite: DOI Citation Formatter
- RDA / Force11 Working Group Output: Use cases and identifier schemes for persistent software source code identification (V1.1)
    - Provides an overview of the current state-of-the-art of the practice of software identification, including use cases and identifier schemes from different academic domains and in industry.

## Software licensing

- ARDC Guide: Research Software Rights Management Guide
- ARDC presentation: Software publishing, licensing, and citation
- ARDC guide: How to make your software citable
- Navigate open source licenses for software: Choosealicense.com and TLDRLegal
- Linux Foundation's SPDX License List

## Publishing software / code availability

- Why share?

    - Article: “Publish your computer code: it is good enough”
    - Article: "Top ten reasons not to share your code and why you should anyway”
- How to share: ARDC guide: How to make your software citable
- Top 10 guides: Top 10 FAIR for research software
- Find a repository: Domain Specific Data Repositories
- Find a repository: Registry of Research Data Repositories (re3data)
- rOpenSci Packages: Development, Maintenance, and Peer Review
- See also the integrations section below
- Software Journals
    - Journal of Open Source Software (JOSS)
    - Journal of Statistical Software (JStatSoft)
    - Journal of Open Research (JORS)

## Apply FAIR to Research Software

- Top 10 guides: Top 10 FAIR for research software
- FAIR4RS reading materials:
    - FAIR4RS Zotero group
    - FAIR4Software reading materials (Research Data Alliance)
    - FAIR for Research Software (FAIR4RS) working group (Research Data Alliance)
- Metadata for Software:
    - Standard: CodeMeta
        - with Multiple crosswalks
    - Standard: Citation File Format (metadata)
    - Howto: Zenodo metadata in a GitHub repository

## Tools/Services/Workflows Discovery

- Biocommons: ToolFinder (national)
- Bioinformatics: Bio.Tools (international)
- Workflows: workflowhub.eu (international)
- Workflows: dockstore.org (international)
- Computational Workbench: CodeOcean
- See subsection Reproducibility

## Maintenance and Preservation

- Platform: Emulation-as-a-Service
- Open Source Software Archive: Software Heritage

- Check data retention policies of Domain specific repositories, most of them will have the mission to preserve scientific digital outputs

## Open Access, Open Research, Open Source

- Services and best practices for OA: CHORUS
  - For Institutions
  - For Researchers
- Open Source:
  - Find open source software
  - Pick an open source license
- University of Melbourne:
  - "Open up your research"
  - "What is Open Research?"
- UQ
  - Open Access
  - Open Access for UQ Research Publications
- CAUL: open research toolkit
- Nature: "Reporting standards and availability of data, materials, code and protocols"
- OA infrastructure: clockss.org (digital preservation archive for scholarly content)

## Institutional Resources

- University of Melbourne:
  - Digital Stewardship Team
  - "Open up your research"
  - "What is Open Research?"
- University of Queensland:
  - Open Access for UQ Research Publications
  - UQ Software Policies and Procedures (only covers software use)

## Learn to use version control

- Software Carpentry: Git for novices
- CSIRO data school: Git with RStudio, based on Software Carpentry materials
- Intersect: Git Training course, based on Software Carpentry materials
- Git for R users: Happy Git and GitHub for the R user
- Other various Git resources for novices and those already using Git

## Reproducibility

- Cloud environment:
  - Binder
  - EaaSI
  - Computational Workbench: CodeOcean
  - Reproducible article (eLife)
- Local environment:
  - For Python: with Anaconda
  - For R: with renv
- Containerisation:
  - ARDC Nectar Research Cloud: container services
  - Docker
  - Kubernetes
  - Containers for HPC: Singularity
  - Standardisation by OCI
- Learning resources:
  - The Carpentries' "Reproducible Computational Environments using Containers" lesson (beta version)
  - The Turing Way
- ReScience C Journal

## Code development infrastructure at institutions in Australia:

- University of NSW, Sydney: GitHub instance
- University of Melbourne: GitLab instance
- Deakin University: GitLab instance
- University of Technology, Sydney:  GitLab instance
- CSIRO: Bitbucket instance
- University of Sydney: GitHub instance (guidance)
- University of Queensland: not yet offering centralised code-hosting, but could be integrated into the UQ Research Data Manager

## Integrations

- Figshare + github
- Figshare + GitHub/GitLab/BitBucket
- Zenodo + GitHub (also ARDC guide: How to make your software citable)
- Dryad + eLife (reproducible article)
- Dryad + GitHub (via Zenodo)
- AARNet Cloudstor + OSF

# The bigger picture

- OECD Recommendation Concerning Access to Research Data from Public Funding
    - The 2021 revision expands the scope to cover not only research data, but also related metadata, as well as bespoke algorithms, workflows, models, and software (including code), which are essential for their interpretation. Australia is a signatory to the Recommendation.
- ARDC National Agenda for Research Software https://bit.ly/rs-agenda
- Open Research and University of Melbourne
- Article: "The challenge and promise of software citation for credit, identification, discovery, and reuse"
- Webinar: Zipkin, Elise, Davis, Kayla, & Stall, Shelley. (2021, September 3). Culture Change: How Studying Monarchs Inspires Transparent and Reproducible Science
- Blog: Enhanced support for software citations on GitHub
- Article: Akhmerov et al. (2019). Raising the Profile of Research Software (1.0.0)
- Nosek et al. 2015 Promoting an Open Research Culture
- The TOP Guidelines:
    - "The TOP Guidelines"
    - Woolston 2020 "Top Factor rates journals on transparency, openness
- Article: The FAIR Guiding Principles for scientific data management and stewardship
- Article: Towards FAIR Principles for Research Software
- Recommendations: FAIR Principles for Research Software (FAIR4RS Principles)
- Draft UNESCO Recommendation on Open Science
- Knowles, Mateen, & Yehudi 2021 "We need to talk about the lack of investment in digital research infrastructure"

# Communities:

- Research Software Engineering Associations:
    - RSE-AUNZ
    - RSE Society (UK)
    - Research Software Engineers International
- Institutional Hacky hours
- Software Sustainability:
    - Software Sustainability Institute (UK) See their guides
    - Research Software Engineer Stories podcast
    - URSSI
    - ARDC Research Software Agenda
- Domain specific communities of practice (a non exhaustive list)
    - ABACBS for bioinformatics - (health and medical, Agriculture, Ancestry, etc)

    - ADACS for astronomy
    - Brainhack Australian chapter
    - Australasian Association for Digital Humanities
    - Geophysics 2030 Project
    - Statistical Society of Australia (SSA)

## AGU (American Geophysical Union)

The AGU (www.agu.org) is an international organisation that supports researchers and practitioners in the Earth and Space Science. The AGU has evolved into a global body with interest groups that cover the full spectrum of the discipline including education, information science, geo-biology, geo-chemistry, planetary science, atmospheric science, ocean science as well as the original "solid Earth" geophysics research areas that the *American Geophysical Union* was built upon.

Geoscience is a broad and cross-disciplinary domain where most active researchers have been trained and educated in non-geoscience schools and bring a diversity of research practices with them. The AGU has evolved to support this broad community and is a leader in integrating data science, collaboration tools, and simulation into research, peer-reviewed publication and education / training. The lessons from the path that the AGU has forged have relevance beyond the Earth Science discipline area. AGU is leading the integration of research software practices at all levels of day-to-day research activity.

- Data & Software Sharing Guidance for Early Career Researchers: AGU Journal Requirements
- Community
- Video: Software as research object
- Talks: "The Paper and the Data"
- Resources: Zenodo list
- Video: EGU Great Debate on Research Software
    - Really interesting insights into the acute concerns of a specific domain
- "Researcher digital presence"

Geosciences is an interesting case study since the "rejuvenation" of the discipline with the development of the theory of plate tectonics following a post-world-war-II explosion in observational data from the oceans and global seismology coincided with the availability of digital computing in the civilian research community. Early computational geophysics research was published purely as supplementary information in research papers with the focus on explaining algorithms, benchmarking and resolution tests, but no public sharing of source code or explicit discussion of licences. Typically, very little value was given to the specific implementation of algorithms and the writing of research code (from scratch) was considered the appropriate apprenticeship for a PhD in computational geoscience. Code sharing became more important as the algorithmic complexity exceeded the scope of a single PhD. Some notable examples from the tectonics community include:

- At Caltech, the introduction of finite element methods in plate / mantle modelling occurred through collaboration with engineering and introduced a culture of code sharing that ultimately resulted in the ConMan and the Citcom code being made available to the community with open-source licences. Parallel versions of these and other codes were also released to the community and this helped cement the view that source code had an intrinsic value.
- The Generic Mapping Tools (GMT) was initially developed by graduate students during their PhD research, and has become a broad community effort in the subsequent four decades.
- The Gplates software is used for the reconstruction of plate motions into geological deep time. It is used by thousands of researchers throughout the geological / tectonics community and has been developed so that it can orchestrate workflows that engage other software tools.

In the early 2000s, a number of initiatives established open-source software development as a core activity in the geosciences and were funded to promulgate good software engineering practices in coding as well as encouraging the adoption of source code revision control and regression / unit testing. These included the geoframeworks initiative and the Computational Infrastructure in Geodynamics organisation, and the Community Surface Dynamics Modeling System (CSDMS) in the US and AuScope in Australia.

# 4. Next activities

The main reason to organise this session was to bring people from different sectors to discuss ways to incentivise and motivate visibility of research software and their authors. We targeted those who might already be giving advice in relation to publishing digital research outcomes and those who are research software authors or support staff. The data discussion in the last decade has shone its benefits and we aim to leverage those efforts for other digital outputs such as research software. As the title suggests, data is only half the battle.

## 4.1. Community

We would like to align with pre-existing national communities, such as, the hacky hour organisers, data/digital librarians, Australian Software Citation Interest Group, and domain specific communities of practice interested to learn and share best practices for research software findability, accessibility, interoperability and reusability. To get involved or for any questions about the BOF or this report email contact@ardc.edu.au with the subject: “Research Software Authors Support”.

Future actions depend on the will or interest of the community, to nurture a community where members feel able to share resources, incentives and processes. To discuss with each other and ask questions about what works well. Some examples of potential future activities are:

1) Continue to generate awareness of existing international, national, regional and institutional policy to support research software authors and their outputs.

2) Articulate how to converge on methods for research software sharing and attribution.

3) Highlight breadth and suitability of nationally / internationally available infrastructure for software registration, development, publishing, discovery, identification, preservation, and re-execution environments. In cases where this infrastructure doesn't exist, this might be an opportunity to perform a gap analysis with collaborators.

4) Community members in general are invited to be involved in promoting best practices of research software development and institutional relevant practices.

What this community can do better together is gather guidance for sharing research software code and promotion of those guidelines. After raising awareness of different identified institutional guidance we can send a coordinated message about best practices across Australian Institutions and to channel international guidance to be customised and adopted as common practice.

The BOF organisers will continue open calls for collaboration via email communications, webinars and/or workshops. This could be beneficial across domains and research institutions, if leaders/representatives from these communities are motivated to take on more responsibility for coordinating the above listed activities.

In addition, for a simpler approach we also created a twitter hashtag #ResearchSoftwareAU and encourage other people to use it, specially to share incentives for sharing research software for the recognition of research software authors and their contribution to research innovation.