






Cloud Solutions for Open Software



Giulia Caldoni
Sara Coppini
March 2023



Table of Contents

01

Context

Reproducibility Crisis

FAIR principles

Software

Reproducibility

02

Research Questions

03

Benchmark

A preliminary step

Cloud platforms for
software reproducibility

Code Ocean

WholeTale



Reproducibility crisis

It has surfaced that a large percentage of studies cannot be reproduced (...) respective studies in psychology and cancer suggest that only 10 to 40% of results are robust in replication.

(Weber, C. F. (2022). Why We Need Open Science. Berlin Exchange Medicine. Retrieved from <https://journal.medicine.berlinexchange.de/pub/swjogoy1>)

NEWS | HEALTH

More than half of high-impact cancer lab studies could not be replicated in controversial analysis

Cancer reproducibility project couldn't assess many papers because of uncooperative authors and other challenges

7 DEC 2021 • 8:50 AM • BY JOCELYN KAISER

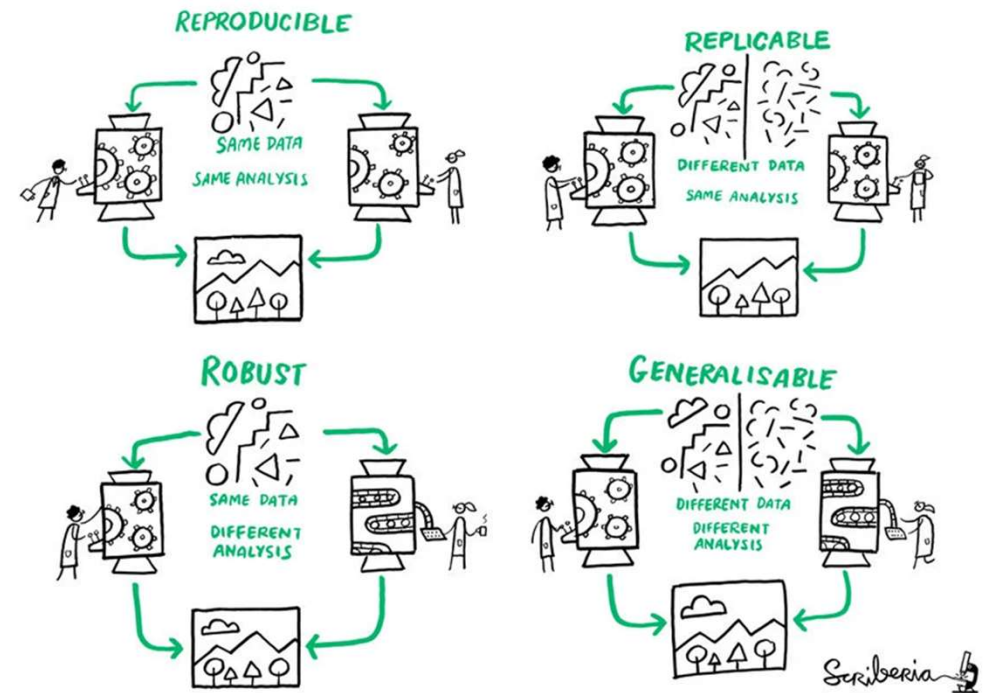
News in focus

IS AI FUELLING A REPRODUCIBILITY CRISIS IN SCIENCE?

'Data leakage' threatens the reliability of machine-learning use across disciplines, researchers warn.

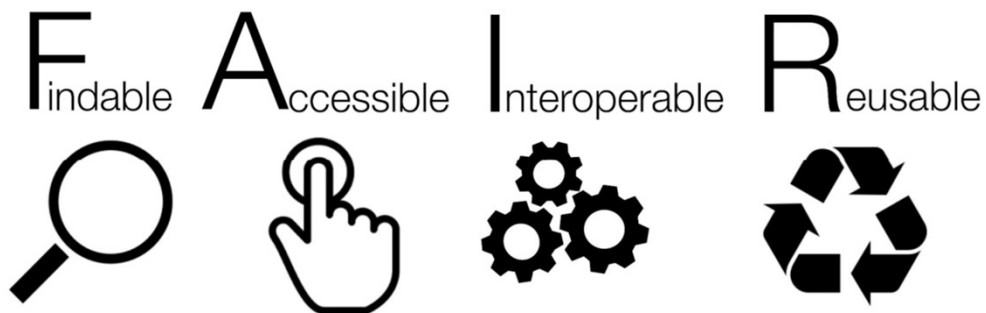
By Elizabeth Gibney

In Atlanta, whose paper² was examined by the pair, says that the field of conflict prediction



Reproducible, replicable... (Image by Scriberia for The Turing Way community, used under a CC-BY license <https://zenodo.org/record/3332808>)

FAIR principles



The FAIR4RS Principles are:

F: Software, and its associated metadata, is easy for both humans and machines to find.
<p>F1. Software is assigned a globally unique and persistent identifier.</p> <ul style="list-style-type: none"> F1.1. Components of the software representing levels of granularity are assigned distinct identifiers. F1.2. Different versions of the software are assigned distinct identifiers. <p>F2. Software is described with rich metadata.</p> <p>F3. Metadata clearly and explicitly include the identifier of the software they describe.</p> <p>F4. Metadata are FAIR, searchable and indexable.</p>
A: Software, and its metadata, is retrievable via standardized protocols.
<p>A1. Software is retrievable by its identifier using a standardized communications protocol.</p> <ul style="list-style-type: none"> A1.1. The protocol is open, free, and universally implementable. A1.2. The protocol allows for an authentication and authorization procedure, where necessary. <p>A2. Metadata are accessible, even when the software is no longer available.</p>
I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.
<p>I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards.</p> <p>I2. Software includes qualified references to other objects.</p>
R: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software).
<p>R1. Software is described with a plurality of accurate and relevant attributes.</p> <ul style="list-style-type: none"> R1.1. Software is given a clear and accessible license. R1.2. Software is associated with detailed provenance. <p>R2. Software includes qualified references to other software.</p> <p>R3. Software meets domain-relevant community standards.</p>


Table 1: The FAIR Principles for Research Software



Software reproducibility (1/5)



Research software: includes both source code and executables used as part of the research process



The trend toward open science increases the pressure on authors to **provide access to the source code and data they used** to compute the results reported in their scientific papers.



Lack of source code along with raw data, and protocols has been described as the **main barrier to computational reproducibility** of published research

(<https://doi.org/10.1371/journal.pone.0251194>).



Even when source code is technically available, **important information about versions, parameters, and runtime environments is often missing** from the scholarly record

(<https://doi.org/10.1002/asi.23538>; <https://doi.org/10.1007/s11192-016-2138-4>).

Software reproducibility (2/5)

A variety of tools have been developed to facilitate computational reproducibility:

- ✓ **Literate programming tools** such as Jupyter notebooks (<https://jupyter.org/>) allow researchers to combine data, code, comments, and outputs (e.g., figures and tables) in a human-readable fashion.
- ✓ **Packaging and containerization platforms** such as ReproZip (<https://www.reprozip.org/>) and Docker (<https://www.docker.com/>) enable the tracking, bundling, and sharing of all of the software libraries and dependencies associated with a research project.
- ✓ **Services like Figshare** (<https://figshare.com/>) and **Zenodo** (<https://zenodo.org/>), via their integration with **GitHub** (<https://github.com/>), allow researchers to deposit, archive, and receive persistent identifiers for their software.
- ✓ Other noteworthy tools such as **workflow management systems** and **online platforms for developing and sharing reproducible methods** allow to document and share computational procedures (e.g., [Galaxy](#) and [Protocols.io](#))



Software reproducibility (3/5)

Journals and publishers must understand and respond to these challenges in the research communities they serve if they wish to support open, reproducible research, and test and implement solutions.

In the Life Sciences:

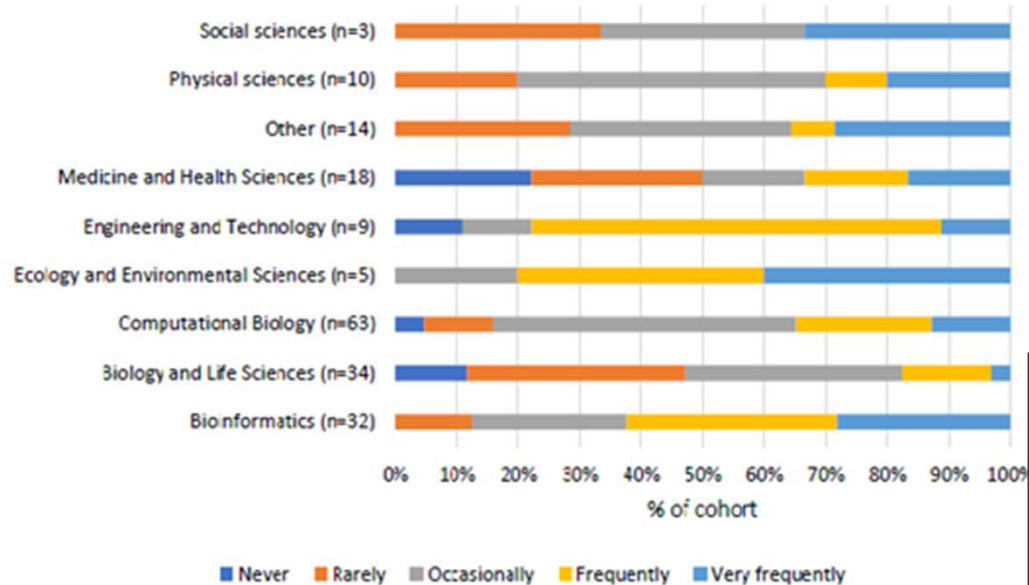
- ✓ *Cell Systems* has a policy that mandates sharing of research data and code as a condition of publication, as *Genome Biology* and *Journal of Royal Society Interface*.
- ✓ *eLife*, together with reproducible document platform, *Stencila*, experimented the publication of Executable Research Articles (ERA; e.g., <https://elifesciences.org/articles/30274/executable>).
- ✓ *Nature Research* journals encourage authors to provide a statement about code availability; *Nature Methods*, and *BMC Bioinformatics*, have trialed offering authors Code Ocean.

Outside of life sciences, sharing of analytic code is commonplace in some areas of the social sciences, such as political science (<https://ajps.org/ajps-verification-policy/>) and economics (<https://www.aeaweb.org/journals/data/data-code-policy>), to support replication or verification of published results.

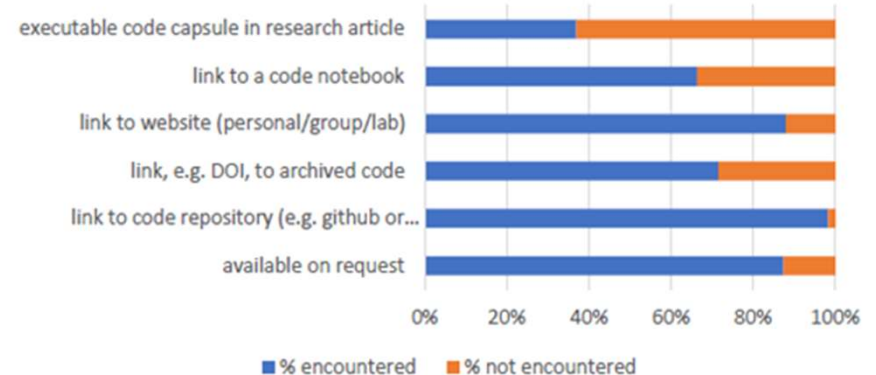
Code Ocean is also fully integrated with IEEE *Xplore*, so readers can discover, browse, run, modify code, and input data to experiment, reproduce, and build on an author's research, all in the cloud, without any other setup or software license.

Software reproducibility (4/5)

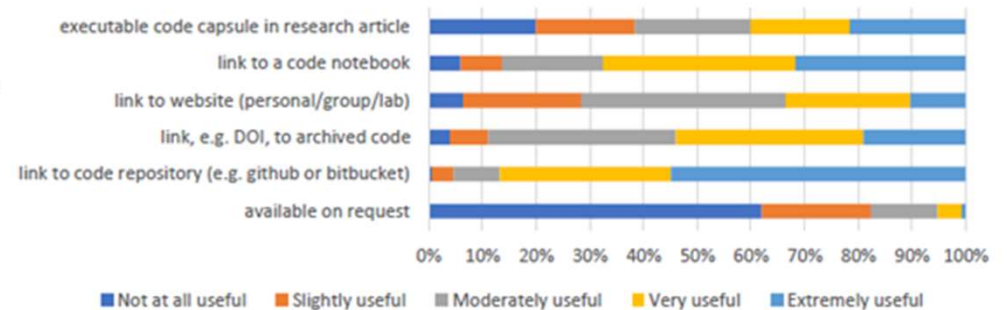
How often do you look at code associated with research articles?



Rates at which different methods of code sharing have been encountered



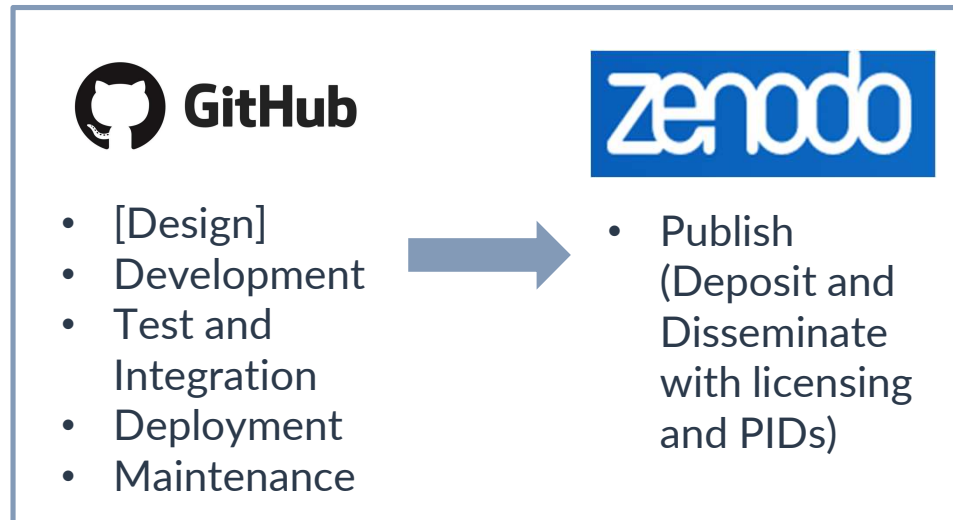
In the last 6 months, how useful did you find the following methods of accessing code? [excluding those who have not encountered the method]



Cadwallader L, Hrynaszkiewicz I. 2022. A survey of researchers' code sharing and code reuse practices, and assessment of interactive notebook prototypes. *PeerJ* 10:e13933 <https://doi.org/10.7717/peerj.13933>

Software reproducibility (5/5)

Current practice



- **But GitHub is a forge** (i.e., a web-based collaborative software platform for developing and sharing computer applications) **thus has no guarantees on sustainability and the long-term preservation of software**
- **In recent years, many forges have been at risk:**
 - 2015: Google Code e Glitorious.org shutdown
 - 2019: bitbucket no longer supports a system technology, the MErcurial VCS sunset, so it suppresses 250,000+ repositories
 - 2022: GitLab plans to delete all projects that have been inactive for a year

- **New possible practice:** relying on **software archives** for depositing and publishing (with licenses and PIDs) such as **Software Heritage**:
 - Thought for long-term preservation of source code
 - Both index and catalog of existing software throughout several online platforms
 - Crypted identification system (SWHID)
 - Archiving contextual elements that allow for code understanding and reconstruction
 - Allow for proper and specific software citation





Research Questions

Data reproducibility is an issue that has been widely discussed and addressed in the last few years. But what about software?








- ✓ What resources are currently available to help researchers ensuring code preservation and reproducibility?
- ✓ What resources meet the needs of researchers working with software and, meanwhile, respond to the FAIR principles?





Benchmark

General features of cloud platforms for software reproducibility

-  **Code Ocean:** Sharing and repository platform with integrated services, including academics ones. Not only to run notebooks or code files to reproduce them. It is structured in 4 sections: capsules, articles, peer reviews and explore. A capsule is composed by code + (optional) data + computational environment.
-  **Binder:** open-source project to launch a public repository from GitHub to run and test your code and download the output results.
-  **Colaboratory:** Cross between a Jupyter Notebook and Google Docs: share, comment and edit notebooks between users, which are stored in Google Drive.
-  **Nextjournal:** Creates copies of existing notebooks to execute code with any package, set environment variables, data sources or export as Docker images to be pulled to run locally or reused in other notebooks.
-  **Kaggle:** interactive and community-driven platform where students and professionals can upload datasets, create notebooks, share ideas in blogs, and participate in competitions.
-  **Deepnote:** project-based notebook platform with database integrations to experiment on a data science project, create a custom environment, have live collaboration, and publish your work.
-  **Whole Tale:** scalable, open-source, multi-user platform to create, publish and execute "tales", i.e., research objects that capture data, code, and the software environment used.



Benchmark

A preliminary step



- **Software for containers visualisation and execution:** software tool that generates ‘containers’ — standardized computational environments that can be shared and reused – mainly locally/remotely. Some examples: [Docker](#), [LXC](#), [OpenVZ](#)
- **Cloud-based services with environment and system specifications** which makes them compatible with or complementary to software for running containers. Researchers can run code in the cloud without needing to *install* more software. They can lock down their software configurations, migrate those environments from laptops to high-performance computing clusters and share them with colleagues. Educators can create and share course materials with students, and journals can improve the reproducibility of results in published articles.



- Most of these cloud-based services are **cloud notebooks**: electronic notebooks (i.e., Jupyter Notebooks) run on the cloud to improve collaboration and reproducibility. Some examples are [Google Colab](#), [Nextjournal](#), [Kaggle](#), [Deepnote](#), [Whole Tale](#), [Hex](#), [Jetbrains Datalore](#), [Noteable](#), [Amazon Sagemaker](#).



Benchmark

Technical features of cloud platforms for software reproducibility

Name	Supported Languages	Affiliations	Integrations	Costs	Collaboration	Set-up
Code Ocean	C/C++, Fortran, Java, Julia, Lua, MATLAB, Python, R, Stata	AACR, Cambridge University Press, Lantern Pharma, IEEE, Nature, Sema4, BMC, Champions Oncology, Voyager Therapeutics, Javelin, DragonFly...	CLOCKSS, Project Jupyter, JupyterLab, R Shiny, Rstudio, Terminal, Amazon	Free limited resources + custom paid plans	Different modes	Fully managed
Binder	Python, Julia, R	Turing Institute, Moore Foundation, 2i2c, UC Berkley, Simula Research Lab	Project Jupyter, Google Cloud, OVH, GESIS Notebooks	Free limited resources	Only on the Jupyter version	Self-hosted
Colaboratory	Python	nd	Project Jupyter, Google (parent), Markdown, GitHub	Free limited unspecified resources + fixed paid plans	Different modes	Fully managed
Nextjournal	Python, R, Julia, Clojure, Tensorflow, Keras, Pytorch, Bash	NASA, ETCHzurich, Carnegie Science, UC Berkley, University of Michigan, UCSB, Swiss Institute of Bioinformatics	Project Jupyter, Markdown, Docker, AWS, GitHub, Google, Clojure, ClojureScript, Datomic, Hashicorp	Free unlimited & limited resources + fixed paid plans	Different modes & real-time	Fully managed
Kaggle	Python, R	nd	Google (parent), Shift, Jupyter Notebook, Rambox, Giskard, Pipedream, Obviously AI	Free limited & unlimited resources	Different modes	Fully managed
Deepnote	Python, R, SQL	Private companies & universities as Oxford, Harvard, Stanford, Cambridge, Caltech	30+ integrations	Free limited resources + custom or fixed paid plans	Different modes & real-time	Fully managed
Whole Tale	Python, R, MATLAB	Universities: Illinois, Chicago, Santa Barbara, Texas, Notre Dame	Github, Slack, Jupyter, Rstudio, STAT A, DataONE, Dataverse, and Zenodo	Free limited resources	Different modes	Self-hosted or fully managed

Benchmark

FAIRness of cloud platforms for software reproducibility

Name	Version control	PID	Licensing	Accessibility (w/ no account)	Privacy	Metadata	Open Source	Long-term Preservation
Code Ocean	Yes, with version state	DOI	Yes, free choice (both for data and software)	View code but not run	Yes, on paid plans	Yes necessarily	No	Yes (CLOCKSS)
Binder	Manual (git)	No (external)	Sort of (external)	View & run code	No	No	Yes	ND
Colaboratory	Sort of ("revision history" or via Github)	No (external)	Sort of (external)	View code but not run	Sort of (manual computations)	No	No	ND
Nextjournal	Yes	DOI	ND	No	Yes	No	No	ND
Kaggle	Yes	DOI only for datasets	Yes, free choice	View and interact but not run	Yes (but US based)	Yes	No	ND
Deepnote	Yes	No (external)	Yes, free choice	View and interact but not run	Yes, certified	No	No	ND
Whole Tale	Yes	No (external)	Yes, free choice	No	ND (TBD)	Yes	Yes	ND

Benchmark

Other noteworthy tools



Workflow management systems: upload data and process them using existing tools.

Examples: [VisTrails](#), [Activiti](#), [Apache ODE](#), [Bizagi](#), [Imixs-Workflow](#), [QuickBase](#), [Pyrus](#). A specific subcategory of WMSs are research data analysis platforms, such as [REANA](#)

Bioinformatics WMSs: [Galaxy](#), [Anduril](#), [CLC bio](#), [Cuneiform](#), [Discovery Net](#), [GenePattern](#), [KNIM](#)



Online Platforms for developing and sharing reproducible methods, such as research protocols describing the background, rationale, objectives, design, methodology of a research, e.g., [Protocols.io](#)



Collaborative educational tools: online tools to share simple code exercises and revise them, e.g., [Codespace.io](#)



Collaborative document (with code) editors and authoring tools: tools to document research workflows and final results with interactive documents. Examples: [Authorea](#), [Manuscripts](#), [eLife](#) [Reproducible Document Stack](#), [O2r](#)



Reproducibility aids for containers execution software: tools to collect research data along with all necessary environment variables and options into a self-contained bundle environment to be executed by software like Docker. An example is [ReproZip](#)



Online development environments and code editors: cloud-based code editors which allow for more flexibility and collaborative uses than local IDEs. Ex. [GitHub Codespaces](#), [Gitpod](#), [DataCamp Workspace](#)



Benchmark



We focused our analysis on the resources that, according to our benchmark:

- ✓ present the most appealing features for coding-researchers;
- ✓ are the most FAIR-oriented.

They were Code Ocean and WholeTale.



Code Ocean

Not just a cloud notebook

- **Capsule:** code, data, metadata, packages, dependencies and all environment specifications
- **Technologies:** Git, Docker, Virtual Private Cloud (VPC) - PaaS, Amazon Web Services (AWS), Jupyter, RStudio, Shiny (R package), SaaS, APIs...
- **Free and custom paid plans:**
 - free plan: 10 compute hours per month for academics, access to published research, git hosting, computational workbench & tools, export of compute capsules, peer review support, 20 GB storage, access to verification services.
 - for paid plans (for publishers and organizations) you need to contact Code Ocean support but **no paid plans for academia** (they "can make allowances for one time extensions in certain situations.").

The screenshot displays the Code Ocean web interface. On the left, a file explorer shows the capsule structure with folders for 'metadata', 'environment', 'code', 'data', and 'results'. The 'environment' folder is selected, showing a 'Dockerfile'. The main panel is titled 'Environment' and shows the selected environment 'R (3.5.3 (deprecated))'. Below this, there's a section for 'Additional Packages' with a table of installed packages and their versions. On the right, a 'Reproducible Run' sidebar shows a timeline of runs, including a 'Published Version 1.0' by Jeffrey Perkel on May 14, 2019.

Environment

R (3.5.3 (deprecated))

R is a language and environment for statistical computing and graphics
Ubuntu 18.04 R

Additional Packages ⓘ

Customize the selected environment with any other packages you need. You can also use these package managers to install other package managers, such as for different languages. Packages will be installed on the next capsule run. [Learn more.](#)

Package Managers	Packages
apt-get ⓘ	libgdal-dev 2.2.3+dfsg-2 libgeos-dev 3.6.2-1build2 libjq-dev 1.5+dfsg-2 libproj-dev 4.9.3-2 libprotobuf-dev 3.0.0-8ubuntu1 libudunits2-dev 2.2.26-1 libv8-dev 3.14.5.0-1ubuntu1 pandoc 1.19.2.4+dfsg1build1 protobuf-compiler 3.0.0-8ubuntu1
Bioconductor ⓘ	
R (CRAN) ⓘ	fields 9.8 geosjonio 0.7.0 leaflet 2.0.0 ncdf4 1.16 raster 2.6-7 rgdal 1.2-20 rgeos 0.4-3 sf 0.6-3 sp 1.2-7
R (GitHub) ⓘ	

Post-Install Script ⓘ

If a package isn't available via the above package managers, use this script to download, extract and install it. Please note: this script should not be used to download data and cannot access any capsule folders. [Learn more.](#)

This capsule has no post-install script.

Reproducible Run

Timeline

Select filter...

Sara Coppini ran Jan 16, 2023 Run 3877009

May 14, 2019 Published Version 1.0 Currently viewing

Author ran May 14, 2019 00:00:09

Published Result

hurricane_sst_v2.html 1.66 MB

London_research_land... 1.25 MB

output 7.65 KB

Jeffrey Perkel committed May 14, 2019

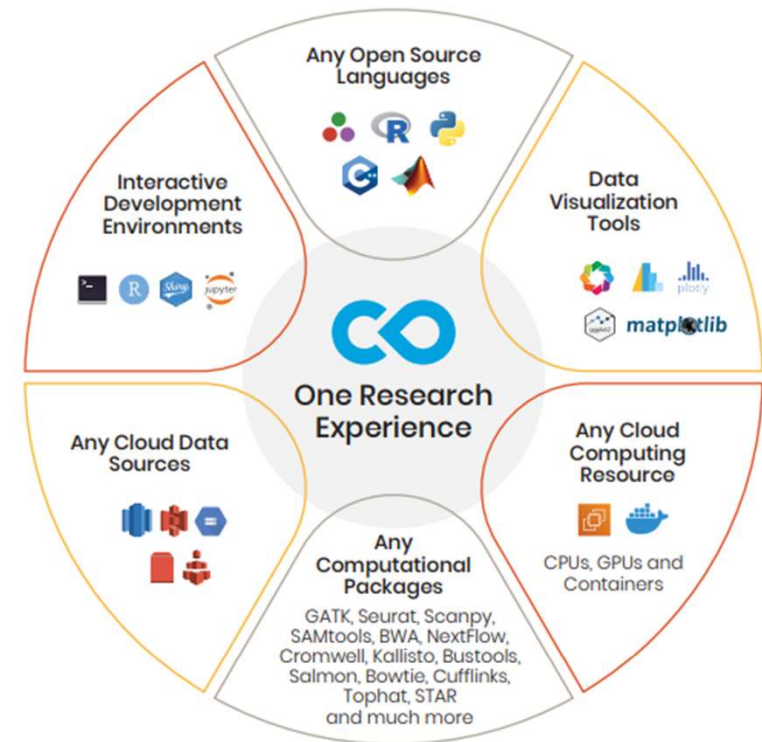
Version 1.0

May 14, 2019 Created Capsule

Code Ocean

Not just a cloud notebook

- **Run computing time:**
 - If the capsule is either *published or duplicated*, the run subtracts minutes from the machine time of the person running the capsule
 - If the capsule is not published but *shared*, the run takes minutes off the machine time of the owner of the capsule
- **Integrated services and tools:**
 - Interactive Development Environments (e.g., Git, Jupyter, JupyterLab, R Shiny, Rstudio e Terminal)
 - Workflow Management Tools (e.g., Cromwell, Snakemake, NextFlow etc.) - **thought for computational biologists**
 - Data Visualization Tools (e.g., matplotlib, plotly etc.)
 - Computational Packages (GATK, Seurat, Scanpy, SAMtools, BWA, Cromwell, Bowtie etc.)
 - Cloud Data Sources for data import and storage
- **4 capsules with authors from the University of Bologna ([a](#), [b](#), [c](#), [d](#))**



Code Ocean

Main features



- Can be integrated directly into online publications as an **interactive widget**;
- Integration with various tools in the **Cloud Workstations**: the code can be executed interactively on **other platforms** line-by-line (e.g., Jupyter) instead of running the whole code file;
- Integration and synchronization with **git-based systems** (e.g., Github);
- Integration with **workflow management tools** like Cromwell, Snakemake, NextFlow;
- Integration with **data analysis apps** to do web-based analysis without using code;

Integrations

- The compute capsules can be **exported**;
- You can view both the "**environment**" file with system specs and the **docker file**;
- Possibility to choose different "**compute resources**" i.e., system characteristics (RAM, cores...);
- **Computational Workbenches** (where you create capsules through the GUI without code);

Flexibility


- You can **share your research asset** with your team or your organization;
- Useful for approaching computation **without specific knowledge** (e.g., automatically resizing computational resources);
- **FAIR design**: navigation by metadata, PIDs, reproducibility, storage

Academic



Code Ocean

Creating and publishing a capsule

- Manual entry of **metadata**, except for some fields (author, title, affiliations...) that autocomplete if you enter the DOI of the publication linked to the capsule
 - Automatic generation of a **yaml file** with metadata, after providing metadata via the user interface
 - Entering a **license** for the data and one for the software, only in case you publish the capsule. You choose the license via drop-down menu in the interface, with the option of uploading a custom license.
 - Creating the capsule **environment**: requirements packages are either entered by hand via the UI or by uploading an external requirements file
 - To run the code, you must create a **"run" file**, specifying files to run
 - Outputs must be saved in the **results** folder, so you may need to modify the code to specify the output path
 - You can install extra packages via postinstall script
 - The **results** are overwritten on each run
- 



Code Ocean

Preservation strategy: CLOCKSS

- Focus on the preservation of **scholarly content** (mainly journals' materials, articles and related datasets)
- CLOCKSS is a **not-for-profit joint venture** developed by **libraries** and **publishers** who guide its management and future (no parent companies): **only joint publishers can submit materials, not individual researchers!**
 - That is why it is not indexed in registries like Re3data
- **Trusted Archive** provided with:
 - certification score by the TRAC audit operated by the Center for Research Libraries;
 - a demonstrated mandate, funding and sustainable business model;
 - a demonstrated track record of preserving academic content;
 - open and transparent documentation and documented agreements, workflows, and processes to ensure long-term access to the repository's contents;
 - succession plan in worst tech scenarios.
- uses **LOCKSS system**
- **Triggered content:** If content that is held in the Archive disappears (or is about to disappear) from the Web, CLOCKSS will “trigger” it for Open Access available with Creative Commons Open Access licenses)

<https://clockss.org/faq/>

<https://clockss.org/about/how-clockss-works/>

Code Ocean

and Publishers

Code Ocean has been **chosen by several publishers** and publishing groups, such as EBSCO Information Services, IEEE, Nature, Lantern Pharma, AACR, De Gruyter, Cambridge University Press.

Example of an article published by the Cambridge University Press, where the **code has been integrated into the text flow** via the Code Ocean widget:
<https://doi.org/10.1017/psrm.2015.83>

Article contents

Abstract

Related Reputations for MCs, Their Party, and Congress?

Observational Data on Spillover Effects

Panel Data

Research Design: Analysis

Results

Discussion

Footnotes

References

Who Gets the Credit? Legislative Responsiveness...

(Daniel M. But...)

Edit Capsule

Files

Core Files

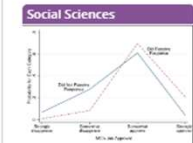
- metadata 1.32 KB
- environment 68 B
- code 13.58 KB
 - CCES Figure 1.R 3.38 KB
 - LICENSE 1.04 KB
 - PSRM-WhoGetsCredit_Replic... 3.17 KB
 - PSRM-WhoGetsCredit_Replic... 5.85 KB
 - run.sh 130 B
- data Manage Datasets 2.64 MB
 - CCES2008_validated_replica... 2.55 MB
 - LICENSE 6.4 KB
 - WhoGetsCredit_LetterWrite... 83.33 KB
- .gitignore 7 B
- results 138.17 KB

Other Files

Open on CODE OCEAN

Metadata

Social Sciences



Who Gets the Credit? Legislative Responsiveness and Evaluations of Members, Parties, and the US Congress

Daniel M. Butler, Christopher F. Karpowitz, Jeremy C. Pope.

This article considers the hypothesis that the positive actions taken by members of Congress (MC) influence citizens' evaluations of them, their party, and Congress as an institution. We begin with a look at the available cross-sectional survey data on contact with legislators and legislator and institutional approval. There legislative responsiveness appears to have a small spillover effect on institutions. However, when we employ a unique panel design that controls for prior levels of opinion and avoids recall bias, we find no evidence of spillover effects. Overall we find that constituents

Code Ocean

and Nature Research journals

- ✓ *Nature Methods*
- ✓ *Nature Biotechnology*
- ✓ *Nature Machine Intelligence*
- ✓ *Nature Computational Science*

Reported to have
partnered with Code
Ocean

(<https://doi.org/10.1038/s41567-019-0434-7>; <https://doi.org/10.1038/s41592-018-0137-5>; <https://doi.org/10.1186/s13059-021-02299-x>;
<https://librarytechnology.org/pr/28006>)

9 – Source Data

We encourage you to provide source data for your figures whenever possible. Full-length, unprocessed gels and blots must be provided as source data for any relevant figures, and should be provided as individual PDF files for each figure containing all supporting blots and/or gels with the linked figure noted directly in the file. Statistics source data should be provided in Excel format, one file for each relevant figure, with the linked figure noted directly in the file. For imaging source data, we encourage deposition to a relevant repository due to size constraints.

Example from author's guidelines for article's submission to
Nature Computational Science:

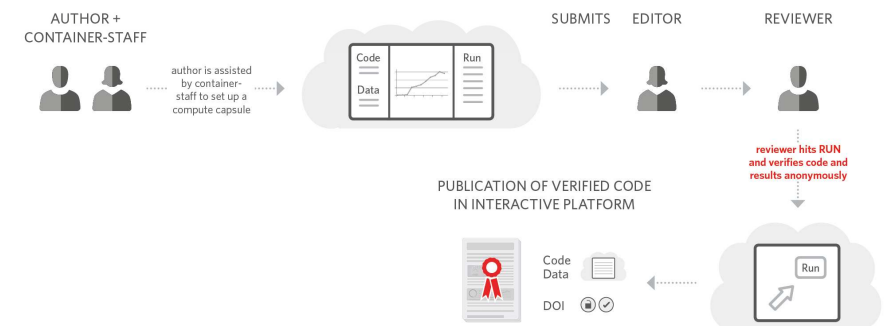
no evident reference to the partnership with Code Ocean

<https://www.nature.com/natcomputsci/submission-guidelines/aip-and-formatting>

1. Traditional peer review of code at the *Nature*-branded journals



2. Executable container-based peer review of code at the *Nature*-branded journals



Code Ocean

and IEEE journals

IEEE Authors: Share the code and data published in your papers

IEEE partnered with Code Ocean to enable authors to enhance the visibility and impact of their research by enabling readers to run the code associated with their articles. Code Ocean is a cloud-based computational reproducibility platform that is fully integrated with IEEE Xplore.

Readers can discover, browse, run, modify code, and input data to experiment, reproduce, and build on an author's research, all in the cloud, without any other setup or software license.

Once the Code Ocean team confirms that everything is working properly, they will have it uploaded to **IEEE Xplore**. Articles with linked code have a *Code & Datasets* tab where readers can run the code without installing or downloading anything.

IEEE authors can join Code Ocean **for free** and publish their data and code in a *compute capsule* linked to the reference article. Automatic metadata population is provided by linking the publication's DOI.

Code Ocean allows authors to increase the visibility and impact of their work by uploading the code associated with an article. Code can be uploaded at any time—during review, acceptance or after publication.

Authors can take advantage of this service so that readers can view and execute code with ease. Authors will also:

- Get credit for all their research output including code and data
- Facilitate reproducibility, replication, and reuse of their research
- Improve discoverability of their code with an assigned DOI
- Preserve their code, data, and the complete environment, so the code always runs

Once the code is published, it will be linked to the associated article in IEEE Xplore.

Learn how to showcase your full research—get started at codeocean.com/signup/ieee

<https://innovate.ieee.org/ieee-code-ocean/#:~:text=Code%20Ocean%20is%20a%20cloud,other%20setup%20or%20software%20license.>

<https://journals.ieeeauthorcenter.ieee.org/create-your-ieee-journal-article/authoring-tools-and-templates/tools-for-ieee-authors/>

WholeTale

- Scalable, open source, web-based, multi-user
- Integrations with Github, Slack, Jupyter, Rstudio, STATA, DataONE, Dataverse, Zenodo
- Creation, publication, and execution of **Tales** - executable research objects that capture data, code, and the complete software environment used to produce research findings
- Whether public or private, you can **run a Tale** only if it's shared with you. Otherwise, you must make a copy to run it (same author, different owner)
- A Tale can be submitted (e.g., **published**) to an external research repository and assigned a **persistent identifier** by the repository – linked with Zenodo, Dataverse and DataONE
- Privacy management still to be defined
- Still a Beta Version ([link](#))

The image displays two screenshots of the WholeTale web interface. The top screenshot shows the 'Tale Dashboard' with tabs for 'My Tales', 'Shared with Me', and 'Public Tales'. A search bar and a 'Create New Tale' button are at the top right. Below, a message states 'You do not have any running Tales.' A grid of four 'Tales' is shown, each with a 'jupyter' logo, a title, a description, and a 'Run Tale' button. The bottom screenshot shows a detailed view of a 'Tale' titled 'em' by Shigettoshi Yokoyama. It includes a 'Current Tasks' section with progress bars for 'Creating Instance' (17%) and 'Building Image'. Below this, a 'Tale is launching, please wait...' message is displayed with a loading spinner.

Bibliography

- A problem shared is a problem halved. *Nature Phys* **15**, 107 (2019). <https://doi.org/10.1038/s41567-019-0434-7>
- Easing the burden of code review. *Nat Methods* **15**, 641 (2018). <https://doi.org/10.1038/s41592-018-0137-5>
- AlNoamany Y, Borghi JA. Towards computational reproducibility: researcher perspectives on the use and sharing of software. *PeerJ Comput Sci.* 2018 Sep 17;4:e163. doi: 10.7717/peerj-cs.163. PMID: 33816816; PMCID: PMC7924683.
- Björn Grüning et al. Practical Computational Reproducibility in the Life Sciences, *Cell Systems*, Volume 6, Issue 6, 2018, Pages 631-635, ISSN 2405-4712, <https://doi.org/10.1016/j.cels.2018.03.014>.
- Cadwallader L, Hrynaszkiewicz I. A survey of researchers' code sharing and code reuse practices, and assessment of interactive notebook prototypes. *PeerJ.* 2022 Aug 22;10:e13933. doi: 10.7717/peerj.13933. PMID: 36032954; PMCID: PMC9406794
- Carlsson, R. (2019, September 4). Computational Reproducibility via Containers in Psychology. <https://doi.org/10.17605/OSF.IO/PS5RU>
- Chue Hong et al, RDA FAIR4RS WG. (2022). FAIR Principles for Research Software (FAIR4RS Principles) (1.0). <https://doi.org/10.15497/RDA00068>
- Howison et al (2015), Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature. (<https://doi.org/10.1002/asi.23538>)
- Hrynaszkiewicz, I., Harney, J., & Cadwallader, L. (2021, April 28). A survey of code sharing practice and policy in computational biology. <https://doi.org/10.31219/osf.io/f73a6>
- Konkol, M., Nüst, D. & Goulier, L. Publishing computational research - a review of infrastructures for reproducible and transparent scholarly communication. *Res Integr Peer Rev* **5**, 10 (2020). <https://doi.org/10.1186/s41073-020-00095-y>
- Pan, X., Yan, E. & Hua, W. Disciplinary differences of software use and impact in scientific literature. *Scientometrics* **109**, 1593–1610 (2016). <https://doi.org/10.1007/s11192-016-2138-4>
- Perkel, J.M. Containers in the cloud. *Nature* **575**, 247-248 (2019); doi: <https://doi.org/10.1038/d41586-019-03366-x>
- Seibold H, Czerny S, Decke S, Dieterle R, Eder T, Fohr S, et al. (2021) A computational reproducibility study of PLOS ONE articles featuring longitudinal data analyses. *PLoS ONE* **16**(6): e0251194. <https://doi.org/10.1371/journal.pone.0251194>
- Stephen R Piccolo, Michael B Frampton, Tools and techniques for computational reproducibility, *GigaScience*, Volume 5, Issue 1, December 2016, s13742–016–0135–4, <https://doi.org/10.1186/s13742-016-0135-4>
- Weber, C. F. (2022). Why We Need Open Science. Berlin Exchange Medicine. Retrieved from <https://journal.medicine.berlinexchange.de/pub/swjogoy1>

Sitography

- <https://innovate.ieee.org/ieee-code-ocean/#:~:text=Code%20Ocean%20is%20a%20cloud,other%20setup%20or%20software%20license>.
- <https://journals.ieeeauthorcenter.ieee.org/create-your-ieee-journal-article/authoring-tools-and-templates/tools-for-ieee-authors/>
- <https://librarytechnology.org/pr/28006>
- <https://www.nature.com/natcomputsci/submission-guidelines/aip-and-formatting>
- <https://blogs.nature.com/ofschemasandmemes/2019/10/09/what-we-have-learnt-testing-container-platforms-for-peer-review-and-publication-of-code>
- Mybinder.org documentation, <https://mybinder.readthedocs.io/en/latest/>
- Code Ocean product sheet, <https://codeocean.com/wp-content/uploads/2020/12/Product-Datasheet.pdf>
- Colaboratory FAQ documentation, <https://research.google.com/colaboratory/faq.html>
- Nextjournal official website, <https://nextjournal.com/>
- Whole Tale documentation, <https://wholetale.readthedocs.io/en/stable/README.html>
- DeepNote documentation, <https://deepnote.com/docs>
- (Weber, C. F. (2022). Why We Need Open Science. Berlin Exchange Medicine. Retrieved from https://journal.medicine.berlinexchange.de/pub/_swjogoy1)
- <https://www.go-fair.org/fair-principles/>
- <https://www.openaire.eu/>
- <https://datasciencenotebook.org/jupyter-notebook-online>