

Implementing Computational Reproducibility in the Whole Tale Environment

Kyle Chard
Computation Institute
University of Chicago
Chicago, IL USA
chard@uchicago.edu

Niall Gaffney
Texas Advanced Computing Center
University of Texas at Austin
Austin, TX USA
ngaffney@tacc.utexas.edu

Matthew B. Jones
National Center for Ecological Analysis and Synthesis
University of California at Santa Barbara
Santa Barbara, CA USA
jones@nceas.ucsb.edu

Kacper Kowalik
NCSA
University of Illinois at Urbana-Champaign
Champaign, IL USA
kowalikk@illinois.edu

Bertram Ludäscher
School of Information Sciences
University of Illinois at Urbana-Champaign
Champaign, IL USA
ludaesch@illinois.edu

Jarek Nabrzyski
Center for Research Computing
University of Notre Dame
South Bend, IN USA
Jaroslaw.Nabrzyski.1@nd.edu

Victoria Stodden[†]
School of Information Sciences
University of Illinois at Urbana-Champaign
Champaign, IL USA
vcs@stodden.net

Ian Taylor
Center for Research Computing
University of Notre Dame
South Bend, IN USA
ian.j.taylor@gmail.com

Matthew J. Turk
School of Information Sciences
University of Illinois at Urbana-Champaign
Champaign, IL USA
mjturk@illinois.edu

Craig Willis
NCSA
University of Illinois at Urbana-Champaign
Champaign, IL USA
willis8@illinois.edu

ABSTRACT

We present and define a structured digital object, called a “Tale,” for the dissemination and publication of computational scientific findings in the scholarly record. The Tale emerges from the NSF funded Whole Tale project (wholetale.org) which is developing a computational environment designed to capture the entire computational pipeline associated with a scientific experiment and thereby enable computational reproducibility. A Tale allows researchers to create and package code, data and information about the workflow and computational environment necessary to support, review, and recreate the computational results reported in published research. The Tale then captures the artifacts and information needed to facilitate understanding, transparency, and execution of the Tale for review and reproducibility at the time of publication.

CCS CONCEPTS

• Information systems~Computing platforms • Information systems~Data federation tools • Information systems~Data replication tools • Information systems~Open source software

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
P-RECS'19, June 24, 2019, Phoenix, AZ, USA.

© 2019 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-6756-1/19/06
DOI: <http://dx.doi.org/10.1145/3322790.3330594>

KEYWORDS

Reproducibility; Cyberinfrastructure; Open Data; Open Code; Computing Environments; Publishing Standards

ACM Reference format: Kyle Chard, Niall Gaffney, Matthew B. Jones, Kacper Kowalik, Bertram Ludäscher, Jarek Nabrzyski, Victoria Stodden, Ian Taylor, Matthew Turk, and Craig Willis. 2019. Implementing Computational Reproducibility in the Whole Tale Environment. In *2nd International Workshop on Practical Reproducible Evaluation of Computer Systems (P-RECS'19)*, June 24, 2019, Phoenix, AZ, USA. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3322790.3330594>.

1 Introduction

The scientific research and discovery process today leverages and relies on computational infrastructure for scientific discovery. We believe that scientific projects will continue to become massively more computing intensive, and research computing will as a necessary consequence become dramatically more transparent [1]. This shift is already occurring through the cyberinfrastructure that supports scientific research, which enables productivity and accelerates discovery [2-7].

The Whole Tale project (<http://wholetale.org>) is a response to these changes and opportunities in computationally and data-enabled research [8]. Whole Tale is a web-based and open source cyberinfrastructure project for generating and publishing reproducible research enabled through the exposure of a new object we call a “Tale,” which is an executable research object that captures data, code, workflow information, and details of the

computing environment used to produce research findings. The Whole Tale platform allows users to interactively create and edit Tales to support re-running of the provided code on the provided data to reproduce results as intended and described by the author.

Whole Tale maintains explicit reference to externally registered data with the ultimate goal of supporting “citation traces” (i.e., systematically citing specific versions of data used in analysis). Whole Tale also supports packaging information about the code, data, and execution environment into an archival format that can be submitted (“published”) to a research repository and assigned a persistent identifier. The purpose is to support re-running the published packages later, possibly by other researchers.

Currently, Whole Tale supports the popular interactive analysis environments RStudio and Jupyter along with various customizations and extensions. The platform is designed to support adding new environments as needed. Infrastructure decisions and the definition of the output Tale object have rested on the precise definition of computational reproducibility we support in the Whole Tale project.

This article discusses our adopted definition of computational reproducibility in the Whole Tale project and what effect this choice has on implementation decisions, in particular our definition of a Tale. We discuss the definition of a Tale, from its design philosophy to its contents and metadata. We close with a discussion of our experience on the importance of defining computational reproducibility in the context of the Whole Tale project.

2 The Need to Define Computational Reproducibility

The first discussion of computational reproducibility of which we are aware is from 1991, when economist Nancy Cartwright published a discussion of replicable and reproducible research and described an approach used by Harry Collins in his work [9]. The approach taken by Collins aligns with the approach of Jon Claerbout dating from 1992 [10-12]. Many fields since have developed definitions of “reproducible” and “replicable,” a discussion we do not address in this article, however we refer the reader to articles that do [13-14].

Within the Whole Tale project we faced a decision around our own implementation of reproducibility. For example, what types of provenance information should we capture to include with Tales and with what level of granularity should the underlying computations be exposed? What aspects of computational reproducibility can be captured and supported automatically and what aspects require user input? How much burden can we ask a researcher to take on to comply with Whole Tale reproducibility demands? Do some scientific tools and software used in the community advance reproducibility more than others? How should this aspect be evaluated from the Whole Tale perspective, if at all? What is the vision of Whole Tale regarding the systematic collection of system level reproducibility information that can be published as part of the Tale?

In addressing those questions within the Whole Tale project we took inspiration from the 2007 concept of the “research compendia” presented as follows: “We introduce the concept of a compendium as both a container for the different elements that make up the document and its computations (i.e. text, code, data...), and as a means for distributing, managing and updating the collection.” [15-16].

3 Defining Computational Reproducibility

At a high level we first distinguish three types of reproducibility following previous work [13]: Empirical Reproducibility: redoing the physical or empirical aspects of an experiment, for example at the bench or in a lab, and achieving the same results as the original experiment; Statistical Reproducibility: ensuring the statistical inference methodology allows the results to generalize to a new sample; and Computational Reproducibility: all necessary data and the computer codes are provided to run the analysis again and recreate the scientific findings (as discussed in the previous section).

We focus on the last of these three in Whole Tale: computational reproducibility (also referred to as “reproducibility” herein). Notice that our focus is targeted to computational claims and computational reproducibility is considered an attribute of those claims. In 2013 a hierarchy of five levels of reproducibility assessments for computational scientific claims was published as the result of an ICERM workshop entitled “Reproducibility in Computational and Experimental Mathematics” (https://icerm.brown.edu/topical_workshops/tw12-5-rcem/) [17-19]. Whole Tale is primarily supporting what is termed in the report “Reviewable Research” and “Replicable Research” and aspires to support “Open or Reproducible Research.” These levels of assessment are defined in the report as follows:

1. Reviewable Research. The descriptions of the research methods can be independently assessed and the results judged credible.
2. Replicable Research. Tools are made available that would allow one to duplicate the results of the research, for example by running the authors’ code to produce the plots shown in the publication.
3. Open or Reproducible Research. Well-documented and fully open code and data that are publicly available that would allow one to (a) fully audit the computational procedure, (b) replicate and also independently reproduce the results of the research, and (c) extend the results or apply the method to new problems.

The American Journal of Political Science (AJPS) Replication and Verification Policy provides a broader description that blends together the three types of reproducibility, and their definition is aimed at analytic results [20]. It states that:

The corresponding author of a manuscript that is accepted for publication in the American Journal of Political Science must provide replication materials that are sufficient to enable interested

researchers to reproduce all of the analytic results that are reported in the text and supporting materials. (see <https://ajps.org/ajps-replication-policy/>)

We provide this definition for context. As an example of its stated goals regarding cyberinfrastructure support for computational reproducibility, the Whole Tale project is currently implementing a curated pipeline for publishing the computational aspects of political science research, based on workflows defined by the Odum Institute at UNC Chapel Hill, with the goal of eventually extending the AJPS model to other journals. We define computational reproducibility for the Whole Tale project as follows:

Whole Tale supports computational reproducibility by enabling researchers to create and package code, data and information about the workflow and computational environment necessary to support review and reproduce results of computational analyses that are reported in published research. Whole Tale implements this definition by supporting explicit citation of externally referenced data, capturing the artifacts and provenance information needed to facilitate understanding, transparency, and execution of the computational processes and workflows used for review and reproducibility at the time of publication.

The term “workflow” here includes information such as inputs and function invocation sequences required to regenerate results. “Results” refers to new scientific claims produced by computational and data-enabled processes. “Published” indicates traditional peer-reviewed publications that communicate scientific findings and can include claims published via preprint archives and newer experimental forms of communication. Whole Tale identifies results as the objects to be reproduced.

4 Implementing the Reproducibility Definition in Whole Tale

We discuss the Tale in detail in this section, including the design decisions, Tale contents, metadata, and examples. We begin with a discussion of the design philosophy behind the Tale, describe the Tale contents including format and metadata, then give examples.

4.1 Tale Design Philosophy

Our approach reflects the belief that the Whole Tale infrastructure should be as familiar and intuitive to researchers as possible, and support as many existing conventions it can. We seek to minimize the burden on the user to understand or modify a Tale. When researchers view the contents of a published Tale, our goal is that they should be able to understand it and potentially modify it or use the contents elsewhere. In defining and implementing a Tale, the Whole Tale project seeks interoperability with other efforts to produce compendia and artifacts that advance computational reproducibility. This takes the form of using standard formats and vocabularies where possible and avoiding creating new definitions or standards

unless doing so provides significant value to the research community. At the same time, the Whole Tale project anticipates interfacing with and supporting many repositories and publication venues and will likely need provider-specific adapters. This is because not all providers support the same features or formats. For example, the DataONE and Dataverse platforms have different base metadata formats during dataset creation and neither support file hierarchies allowed by others such as Zenodo.

There are established communities around the development of specific technologies that create compendia for reproducibility and artifact sharing, and we implement conventions that occur outside the Whole Tale project for organizing code, data, metadata, provenance information in workflows. Whole Tale itself is not a repository and as such all publication occurs through the export of Tales from Whole Tale to external resources for long term stewardship and archiving. We rely on external repositories for archival formats and preservation.

4.2 Tale Contents

A Tale is an abstract container for data, code, metadata, and computational environments and their dependencies. Data may be included by copy or by reference, which is an important feature of Whole Tale. Table 1 lists the information included as part of a Tale.

Authors	List of individual Tale authors
Creators	Tale Creators (may be different than authors)
Title	Title of the Tale
Description	Description of the Tale
Categories	List of subject categories (keywords)
Illustration	Illustration for the Whole Tale browse page
Create Date	Date the Tale was created
Update Date	Date the Tale was last updated
License	License selected by the user
Environment	Computational environment information
Workspace	Code/scripts, workflow, narrative, documentation, data, results
External data	Data by reference to external source
Identifier	Persistent identifier for published Tale

Table 1. Objects that comprise the Tale

4.3 Tale Metadata

The metadata that Whole Tale imbues on the Tale is crucial for future archiving and reproducibility. It does, in fact, describe the Tale itself. We currently support two types of metadata:

- Standard descriptive metadata: This includes common attributes such as those provided by DataCite, Dublin Core or schema.org and supported by most repositories.
- System metadata: Configuration information and metadata required by the Whole Tale system, for example the associated base environment

A manifest.json file is included with the Tale that contains the basic Tale metadata as well as a resource map that conforms to the OAI-ORE metadata specification (an example follows). Following our design philosophy we chose to use the BagIt-RO format specification and <https://w3id.org/bundle> to allow us to directly support both hierarchical (Zip, BagIt) and non-hierarchical (DataONE, Dataverse) targets with a single format [21]. Table 2 shows the key:value pairs for each type of metadata associated with a Tale.

Authors	schema:author
Creator	pav:createdBy
Title	schema:name
Description	schema:description
Categories	schema:category
Identifier	schema:identifier
Illustration	schema:image
Create Date	pav:createdOn
License	schema:license + generated file
Environment	See environment.json e.g. [22]
Version	schema:version
Datasets	https://w3id.org/bundle/context
Workspace	https://w3id.org/bundle/context

Table 2. Tale object metadata descriptions

An example of a manifest.json file, implementing the descriptions in Table 2, is available [22]. Figure 1 illustrates the Tale creation process. A researcher begins by uploading or registering data via persistent identifier. A Tale is created by specifying its title and selecting the base computational environment (e.g. Jupyter or RStudio). The Tale runs as a Docker container with the user's workspace available read-write and any externally registered data mounted read-only. The user can specify additional descriptive metadata beyond that from the data and current Tale attributes. The workspace does not prescribe an organizational structure. A typical workspace contains code and scripts, workflows, narrative, documentation, data, results, and environment configuration files (via repo2docker). The user can execute their Tale via the Whole Tale platform or export and run it locally. Once published to an archival repository, the Tale receives a persistent identifier

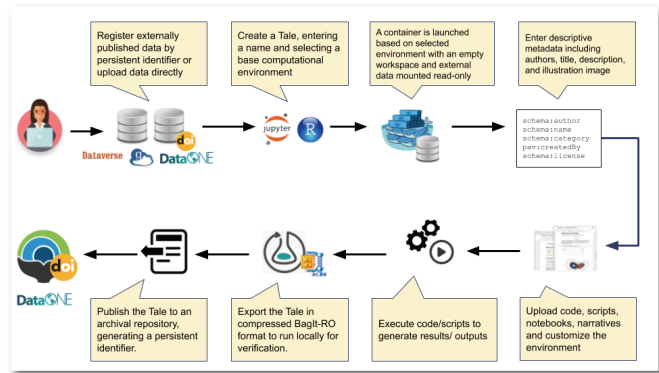


Figure 1: End to end Tale creation workflow

from that repository. Recommended filesystem structure, specific workflow models, and automatic tracking of provenance are currently in planning. In the Whole Tale project, we endeavor to adopt and extend existing tools and best practices.

4.4 An Example Tale

The Whole Tale platform will support publishing to the DataONE network in v0.7, expected to be released in May 2019. An early example of a Tale that predates this feature is that of Ren et al. [23]. Their research on the discovery of metallic glasses using machine learning methods published in *Science Advances* includes a link to the GitHub repository containing the code used in their study (<https://github.com/fang-ren/Discover MG CoVZr>) and data was published to the Materials Data Facility (MDF) [24]. The GitHub repository includes a link to the Tale in the Whole Tale platform that allows readers to quickly access a Jupyter environment with the code, data, and software environment available for reproducibility. In the future, published Tales will receive digital object identifiers (DOIs) as assigned by repositories and be accessible directly from the publication.

4.5 An Example of Publishing to Repositories

The Whole Tale platform is intended to support the creation, publication, and re-execution of Tales. As noted above, Whole Tale is not a repository and relies on integration with existing research data archives for persistent identifiers and long-term preservation. The publication process may require the transformation of the Tale, exported as a BagIt-RO archive, to repository-specific formats. In the simplest case, the archive can be directly published as-is. However, to take advantage of many of the advanced features of repositories including discovery and provenance display, in most cases it is desirable to translate the Tale package. Our initial integration with the DataONE network includes the transformation of the manifest.json to Ecological Markup Language (EML) and Object Reuse and Exchange (ORE) standards.

4.6 The Tale as a Standardized Research Object

Although not stored on Whole Tale after publication, Tales emerging from the Whole Tale platform are intended to be persistent and enable computational reproducibility to the greatest extent possible. As systems change over time executability may no longer be possible, but inspectability will remain. We took inspiration for the Tale's "compendia" approach from the definition of the "Research Compendia" in 2004 [15, 25] and from the implementation persistently linking the underlying data and code to the publication as presented in [26]. Other efforts related to the packaging and publishing of digital scholarly objects are ReproZip which gathers and bundles dependencies for command line executions [27]. Occam is a system that creates "nodes" from objects such as code, and allows a user to specify how to connect nodes to reproduce executions [28]. Popper provides a framework to automatically execute and describe computational experiments [29]. The

Collective Knowledge (CK) framework gathers research objects with unique IDs and metadata in the JSON format [30]. CodeOcean creates exportable “capsules” that contain digital objects needed to regenerate computational results (see <http://codeocean.com>). The Common Workflow Language (CWL) is leveraged to assemble bioinformatics tools into reproducible pipelines in [31]. Sciunits are self-contained bundles guaranteed to re-execute regardless of deployment, and targeted at scientific experiments [32]. Sumatra captures execution dependencies and permits the user to design reproducible workflows for python scripts [33]. Discussions of research object publication occurs at <http://www.researchobject.org>. Interoperability between these different compendia is a desirable goal.

5 Why We Found Defining Computational Reproducibility to be Important

Computation has now emerged as central to the research enterprise, and with it comes the need to enable transparency and inspectability into the report of scientific discoveries. Reproducibility is a fundamental concept to scientific discovery, giving the community important information about the likelihood a result is, in fact, true. Enabling transparency into the computational steps and processes, including data, inputs, and parameters, that underlie a scientific claim allows researchers to make the case to the community that they have exerted appropriate effort to ensure an error is sufficiently rooted out of the discovery process and the corresponding claim. In short, we cannot understand or assess claims presented from computationally opaque processes. Scripting the computational steps is important to expose the implementation of the reasoning behind a proposed claim [34-35]. We seek to meet this standard in the Whole Tale project with the Tale specification.

Researchers, scientists, and others can create tales that allow them to share their findings with the community in such a way that they are computationally reproducible, re-usable, and extensible. This permits interested community members to inspect how the results were generated, including computational decisions made along the entire discovery pipeline, permitting the effective comparison of scientific claims. Review of scientific results during the publication process can be facilitated through access to the Tale, as reviewers can have a (possibly private) Tale shared with them when the manuscript is sent out for review. New claims can be reproduced and verified by reviewers. Researchers can both re-use components of published Tales, perhaps with new methods or data, and extend the research in a new Tale to potentially develop new findings. Publication to repositories and discoverability via repositories is part of the reproducibility vision of Whole Tale. Persistent unique IDs such as Document Objects Identifiers (DOIs) are assigned by repositories, providing an identifier for the Tale itself, and identifiers for objects contained within the Tale to facilitate citation and reuse.

The Tale is intended to interoperate with other existing and evolving standards in the community. Binder for example creates

a computational environment that supports executability and bundling of notebook-based codes for reproducibility [36]. A principle of the Binder ecosystem is to enable researchers to operate with familiar tools and conventions, so instead of requiring a Dockerfile, researchers provide a requirements.txt file to install Python packages. This avoids the creation of a binder.yaml that requires researchers to learn a new convention.

Tales can be exported from Whole Tale, meaning that a researcher can remove their work from Whole Tale as they so desire. Whole Tale does not assert any ownership over code, data, or other information created by the researcher using Whole Tale. The ability to verify and examine computational steps enables debugging and comparisons between different published efforts. We have partially implemented a feature that will allow users to designate the “entrypoint” of a Tale (e.g. a run.sh script or notebook) that shows how to reproduce Tale findings.

6 Conclusions and Future Work

We have presented the definition of the “Tale” as a compendium of digital artifacts assembled to permit computational reproducibility of scientific findings. In the course of doing so we articulated the definition of reproducibility used in the Whole Tale project to define the Tale. Future work includes incorporating intermediary objects, including derived or output data objects produced in the course of the work, in the Tale. Tales could be extended to include other artifacts and information relevant to interpretability and re-execution, such as documentation giving detailed explanations of parameter tuning methods employed or information on how to use the workflow accurately. Future work also includes developing tests for the computations in the Tale. We also plan to implement a concept of validation to make it possible to run a Tale in a “clean” state to confirm that it behaves as expected before publication e.g. review.

Tales are designed to be executable within the Whole Tale environment or on other systems. Tales rely on Docker containerization to bundle objects with computational environment information. Expectations should remain reasonable regarding executability on different systems and especially over time as software and hardware evolve. We believe that exposure of the computational steps and data is useful, even when executability fails. An overarching and challenging goal is to achieve interoperability with other research compendia developed by the community and discussed in Section 4.6.

Using hierarchical DOI assignment for the objects comprising a Tale would reflect the inseparable and inextricable relationships between digital scholarly objects, for example the data or code that are associated with a published claim, and enable discoverability of related objects. Reproducibility badging standards discussions are underway (see e.g. <https://www.niso.org/standards-committees/reproducibility-badging>).

ACKNOWLEDGMENTS

This work is supported by National Science Foundation Award OAC-1541450. We thank the reviewers for helpful comments.

REFERENCES

- [1] H. Monajemi, D. L. Donoho and V. Stodden (2016). Making massive computational experiments painless. *IEEE International Conference on Big Data (Big Data)*, Washington, DC, 2016, pp. 2368-2373. doi: 10.1109/BigData.2016.7840870
- [2] V. Stodden, F. Leisch, R. D. Peng (eds.) (2014). *Implementing Reproducible Research*. Boca Raton, FL: Chapman & Hall/CRC The R Series.
- [3] I. Jimenez, M. Sevilla, N. Watkins, C. Maltzahn, J. Lofstead, K. Mohror, A. Arpac-Dusseau and R. Arpac-Dusseau. (2017). The Popper Convention: Making Reproducible Systems Evaluation Practical. *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 1561–70.
- [4] I. Jimenez, M. Sevilla, N. Watkins, C. Maltzahn, J. Lofstead, K. Mohror, A. Arpac-Dusseau and R. Arpac-Dusseau. (2016). Standing on the Shoulders of Giants by Managing Scientific Experiments Like Software, *USENIX*. 41(4).
- [5] M. McLennan, and R. Kennell. (2010). HUBzero: A Platform for Dissemination and Collaboration in Computational Science and Engineering. *Computing in Science & Engineering*. 12(2). 10.1109/MCSE.2010.41
- [6] N. Wilkins-Diehr. (2007), Special Issue: Science Gateways—Common Community Interfaces to Grid Resources. *Concurrency Computat.: Pract. Exper.*, 19: 743-749. doi:10.1002/cpe.1098
- [7] M. Dahan, J. Towns, T. Cockerill, I Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. Peterson, R. Roskies, J. Scott, & N. Wilkins-Diehr. (2014). XSEDE: Accelerating scientific discovery. *Computing in Science and Engineering*. 16. 62-74. 10.1109/MCSE.2014.80.
- [8] A. Brinckman, K. Chard, N. Gaffney, M. Hategan, M. B. Jones, K.Kowalik, S. Kulasekaran, B. Ludaescher, B. D. Mecum, J. Nabrzyski, V. Stodden, I. J. Taylor, M. J. Turk, K. Turner. (2019). Computing environments for reproducibility: Capturing the "Whole Tale". *Future Generation Comp. Syst.* 94. pp. 854-867.
- [9] N. Cartwright (1991), Replicability, Reproducibility, and Robustness: Comments on Harry Collins, *History of Political Economy*, 23(1). pp. 143-155.
- [10] J. F. Claerbout and M. Karrenbach. (1992). Electronic documents give reproducible research a new meaning. *SEG Technical Program Expanded Abstracts*. pp. 601-604. 10.1190/1.1822162
- [11] J. B. Buckheit and D. L. Donoho. (1995) WaveLab and Reproducible Research. In: Antoniadis A., Oppenheim G. (eds) *Wavelets and Statistics*. Lecture Notes in Statistics, vol 103. Springer, New York, NY.
- [12] R. D. Peng. (2011). Reproducible Research in Computational Science. *Science*. 334(6060). pp. 1226-1227. 10.1126/science.1213847
- [13] V. Stodden. (2013). Resolving Irreproducibility in Empirical and Computational Research. *IMS Bull. Online*. <http://bit.ly/BullIMStat2013>.
- [14] L. A. Barba. (2018). Terminologies for Reproducible Research. *CoRR*. abs/1802.03311.
- [15] R. Gentleman and D. Temple Lang. (2007). Statistical Analyses and Reproducible Research. *Journal of Computational and Graphical Statistics*. 16(1). pp. 1-23.
- [16] V. Stodden and S. Miguez. 2014. Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research. *Journal of Open Research Software*, 2(1), p.e21. DOI: <http://doi.org/10.5334/jors.av>
- [17] V. Stodden, D. Bailey and J. Borwein (2013). Set the Default to 'Open'. *Notices of the AMS*.
- [18] V. Stodden, D. Bailey and J. Borwein (2013). "Setting the Default to Reproducible" in *Computational Science Research*. *SIAM News*.
- [19] ICERM Reproducibility Workshop Report. (2012). Setting the Default to Reproducible. https://icerm.brown.edu/topical_workshops/tw12-5-rcem/icerm_report.pdf
- [20] T. Christian, S. Lafferty-Hess, W. G. Jacoby, and T. Carsey. (2018). Operationalizing the Replication Standard. *International Journal of Digital Curation*. 13(1). DOI: <https://doi.org/10.2218/ijdc.v13i1.555>
- [21] K. Chard, M. D'Arcy, B. Heavner, I. Foster, C. Kesselman, R. Madduri, A. Rodriguez, S. Soiland-Reyes, C. Goble, K. Clark, E. W. Deutsch, I. Dinov, N. Price, A. Toga. (2016). I'll take that to go: Big data bags and minimal identifiers for exchange of large complex datasets. *IEEE International Conference on Big Data*.
- [22] C. Willis. (2019). Whole Tale Example "Water Tale". DOI: 10.5281/zenodo.2641314. <https://zenodo.org/record/2641314#.XLVCCes3nOQ>
- [23] F. Ren, L. Ward, T. Williams, K. J. Laws, C. Wolverson, J. Hattrick-Simpers and A. Mehta. (2018). Accelerated discovery of metallic glasses through iteration of machine learning and high-throughput experiments. *Science*. 4(4). DOI: 10.1126/sciadv.aq1566
- [24] B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, and I. Foster. (2016). The Materials Data Facility: Data services to advance materials science research. *JOM*. 68(8). pp. 2045-2052.
- [25] Gentleman, Robert and Temple Lang, Duncan, (2004). "Statistical Analyses and Reproducible Research" *Bioconductor Project Working Papers*. Working Paper 2. <https://biostats.bepress.com/bioconductor/paper2>
- [26] V. Stodden, S. Miguez, J. Seiler. (2015). "ResearchCompendia.org: Cyberinfrastructure for Reproducibility and Collaboration in Computational Science." *Computing in Science & Engineering*, 17(1). 10.1109/MCSE.2015.18
- [27] F. Chirigati, D. Shasha and J. Freire, (2013). ReproZip: Using Provenance to Support Computational Reproducibility, 5th *USENIX Workshop on the Theory and Practice of Provenance*.
- [28] L. Oliveira, D. Wilkinson, D. Mossé, and B. Childers. (2018). Supporting Long-term Reproducible Software Execution. In *Proceedings of the First International Workshop on Practical Reproducible Evaluation of Computer Systems (P-RECS'18)*. ACM, New York, NY, USA, Article 6, 6 pages. DOI: <https://doi.org/10.1145/3214239.3214245>
- [29] I. Jimenez, M. Sevilla, N. Watkins, C. Maltzahn, J. Lofstead, K. Mohror, A. Arpac-Dusseau and R. Arpac-Dusseau. (2017). The Popper Convention: Making Reproducible Systems Evaluation Practical. *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 1561–70. <https://doi.org/10.1109/IPDPSW.2017.157>.
- [30] G. Fursin, A. Lokhmotov and E. Plowman. (2016). Collective Knowledge: Towards R&D Sustainability. *Proceedings of the Conference on Design, Automation and Test in Europe (DATE'16)*.
- [31] B. Grüning, J. Chilton, J. Köster, R. Dale, N. Soranzo, M. van den Beek, J. Goecks, R. Backofen, A. Nekrutenko and J. Taylor. (2018). Practical computational reproducibility in the life sciences. *Cell Systems* 6.
- [32] Z. Yuan, D. Hai Ton That, S. Kothari, G. Fils, and T. Malik. (2018). Utilizing Provenance in Reusable Research Objects. *Informatics*, 5(1), 14; <https://doi.org/10.3390/informatics5010014>
- [33] A. P Davison, M Mattioni, D Samarkanov, B Tele'nczuk. (2014). Sumatra: A Toolkit for Reproducible Research. In *Implementing Reproducible Research*, Eds: Stodden, V and Leisch, F and and Chapman, R D Peng, pp.57-79
- [34] L. A. M. C. Carvalho, K. Belhajjame and C. B. Medeiros, "Converting scripts into reproducible workflow research objects," 2016 *IEEE 12th International Conference on e-Science (e-Science)*, Baltimore, MD, 2016, pp. 71-80. doi: 10.1109/eScience.2016.7870887
- [35] Sandve GK, Nekrutenko A, Taylor J, Hovig E (2013) Ten Simple Rules for Reproducible Computational Research. *PLoS Comput Biol* 9(10): e1003285. <https://doi.org/10.1371/journal.pcbi.1003285>
- [36] Jupyter et al., (2018). Binder 2.0 - Reproducible, Interactive, Sharable Environments for Science at Scale." *Proceedings of the 17th Python in Science Conference*. doi:10.25080/Majora-4af1f417-011