Patterns

CellPress
OPEN ACCESS

Review

The role of metadata in reproducible computational research

Jeremy Leipzig,[1,*] Daniel Nüst,[2] Charles Tapley Hoyt,[3] Karthik Ram,[4] and Jane Greenberg[1]
[1]Metadata Research Center, College of Computing and Informatics, Drexel University, Philadelphia, PA, USA
[2]Institute for Geoinformatics, University of Münster, Münster, Germany
[3]Laboratory of Systems Pharmacology, Harvard Medical School, Boston, MA, USA
[4]Berkeley Institute for Data Science, University of California, Berkeley, Berkeley, CA, USA
*Correspondence: jnl47@drexel.edu

https://doi.org/10.1016/j.patter.2021.100322

THE BIGGER PICTURE A recent confluence of technologies has enabled scientists to effectively transfer runnable analyses, addressing a long-standing challenge of reproducible research. The implementation of reproducible research for *in silico* analyses requires extensive metadata to describe both scientific concepts and the underlying computing environment. This review covers the wide range of metadata standards relevant to reproducible computational research across an "analytic stack" consisting of input data, tools, reports, pipelines, and publications. Legacy and cutting-edge metadata support a wide range of data annotations, analytic approaches, and interpretation across virtually all scientific disciplines. This review is designed to bridge the metadata and reproducible research communities. We identify competing approaches of embedded and connected metadata, discuss gaps, and make recommendations with implications for the future of journals and peer review.

SUMMARY

Reproducible computational research (RCR) is the keystone of the scientific method for *in silico* analyses, packaging the transformation of raw data to published results. In addition to its role in research integrity, improving the reproducibility of scientific studies can accelerate evaluation and reuse. This potential and wide support for the FAIR principles have motivated interest in metadata standards supporting reproducibility. Metadata provide context and provenance to raw data and methods and are essential to both discovery and validation. Despite this shared connection with scientific data, few studies have explicitly described how metadata enable reproducible computational research. This review employs a functional content analysis to identify metadata standards that support reproducibility across an analytic stack consisting of input data, tools, notebooks, pipelines, and publications. Our review provides background context, explores gaps, and discovers component trends of embeddedness and methodology weight from which we derive recommendations for future work.

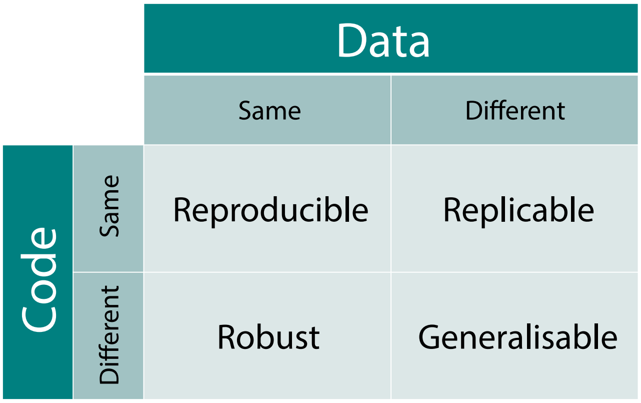
INTRODUCTION

Digital technology and computing have transformed the scientific enterprise. As evidence, many scientific workflows and methods have become fully digital, from the problem scoping stage and data collection tasks to analyses, reporting, storage, and preservation. Another key factor includes federal[1] and institutional[2,3] recommendations and mandates to build a sustainable research infrastructure, to support FAIR principles,[4] and reproducible computational research (RCR). Metadata have emerged as a crucial component, supporting these advances, with standards supporting the research life cycle. Reflective of change, there have been many case studies on reproducibility,[5] although few studies have systematically examined the role of metadata in supporting computational reproducibility. Our aim in this work was to review metadata developments that are directly applicable to computational reproducibility, identify gaps, and recommend further steps involving metadata toward building a more robust infrastructure. To lay the groundwork for these recommendations, we first review reproducible computational research and metadata, examine how they relate across different stages of an analysis, and discuss what common trends emerge from this approach.

Intended audience

This review is designed primarily to bridge the metadata and reproducible research communities. The practitioners working

This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).



		Data	
		Same	Different
Code	Same	Reproducible	Replicable
	Different	Robust	Generalisable

Figure 1. Whitaker's matrix of reproducibility
Whitaker's matrix of reproducibility;[10] made available under the Creative Commons Attribution license (CC-BY 4.0).

in this area may be considered information scientists or data engineers working in the life, physical, and social sciences. Those readers most interested in the representation of scientific data and results will find sections on input and publication most relevant, while those most closely aligned with analysis and data engineering may be more interested in the sections on tools, reports, and pipelines. During the development of this article, it became evident that many important efforts that could be useful and applicable to other domains will wither in isolation if not discovered by a wider audience. Furthermore, many areas of research homologous are not identified as such simply due to differences in the use of terminology. Though much of the battle ground of reproducibility has involved the fields of bioinformatics and psychology, these are by no means the only affected areas. It should be mentioned that while the reproducibility crisis has played out on a public stage involving high-profile papers and journals and is often connected to challenges in peer review processes, the home front of reproducibility is borne by individuals working in smaller settings who need to reproduce analyses written by immediate colleagues, or even themselves.

Reproducible computational research

"Reproducible research" is an umbrella term that encompasses many forms of scientific quality, from generalizability of underlying scientific truth, exact recreation of an experiment with or without communicating intent, to the open sharing of analysis for reuse. Specific to computational facets of scientific research, RCR[6] encompasses all aspects of *in silico* analyses, from the propagation of raw data collected from the wet lab, field, or instrumentation, through intermediate data structures, computational hardware, to open code and statistical analysis, and finally publication. Here, our emphasis is on the scholarly record with results reported in a journal article, conference proceeding, white paper, or report, as a final reporting; although we clearly recognize the importance of reproducibility and full scope of scientific output including data, software, tools, and even data papers.[7] Reproducible research points to several underlying concepts of scientific validity – terms that should be unpacked to be understood. Stodden et al.[8] devised a five-level hierarchy of research, classifying it as reviewable, replicable, confirmable, auditable, and open or reproducible. Whitaker[9] described an analysis as "reproducible" in the narrow sense that a user can produce identical results provided the data and code from the original, and "generalizable" if it produces similar results when both data are swapped out for similar data ("replicability"), and if underlying code is swapped out with comparable replacements ("robustness") (Figure 1).

While these terms may confuse those new to reproducibility, a review by Barba disentangled the terminology while providing a historical context of the field.[11] One major conflicted use of terms (reproducible/replicable) has since then been harmonized.[12] A wider perspective places reproducibility as a first-order benefit of applying FAIR principles: Findability, Accessibility, Interoperability, and Reusability. In the next sections, we engage reproducibility in the general sense and use "narrow-sense" to refer to the same data, same code condition.

Reproducibility crisis

The scientific community's challenge with irreproducibility in research has been extensively documented.[13] Two events in the life sciences stand as watershed moments in this crisis: the publication of manipulated and falsified predictive cancer therapeutic signatures by a biomedical researcher at Duke and subsequent forensic investigation by Keith Baggerly and David Coombes,[14] and a review by scientists at Amgen who could replicate the results of only 6 of 53 cancer studies.[15] These events involved different aspects: poor data structures and missing protocols, respectively. Together with related studies,[16] they underscore recurring reproducibility problems due to a lack of detailed methods, missing controls, and other protocol failures in inadequate understanding or misuse of statistics, including inappropriate statistical tests and/or misinterpretation of results, which also plays a recurring role in irreproducibility.[17] Regardless of intent, these activities fall under the umbrella term of "questionable research practices." It bears speculation whether these types of incidents are more likely to occur in novel statistical or computational approaches compared with conventional ones. Subsequent surveys of researchers[13] have identified selective reporting, while theory papers[18] have emphasized the insidious combination of underpowered designs and publication bias, essentially a multiple testing problem on a global scale. We contend that metadata have an undervalued role to play in addressing all of these issues and to shift the narrative from a crisis to opportunities.[19]

In the wake of this newfound interest in reproducibility, both the variety and volume of related case studies increased after 2015 (Figure 2). Likert-style surveys and high-level publication-based censuses (see Figure 3) in which authors tabulate data or code availability are most prevalent. In addition, low-level reproductions, in which code is executed, replications in which new data are collected and used, tests of robustness in which new tools or methods are used, and refactors to best practices are also becoming more popular. While the life sciences have generated more than half of these case studies, areas of the social and physical sciences are increasingly the subjects of important reproduction and replication efforts. These case studies have provided the best source of empirical data for understanding reproducibility and will likely continue to be valuable for evaluating the solutions we review in the next sections.

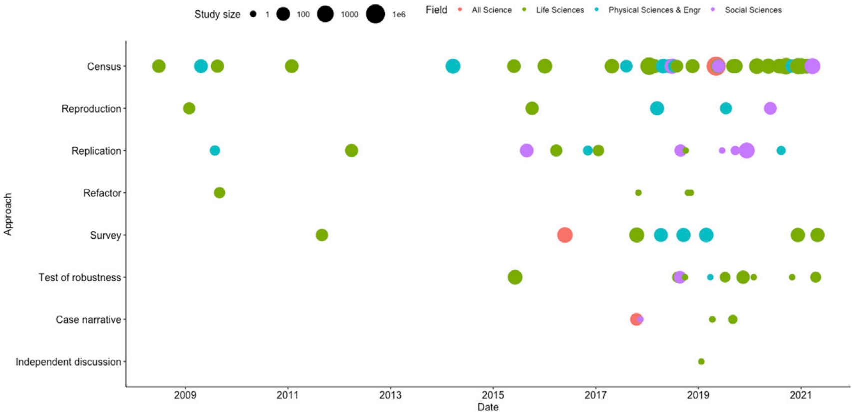


Figure 2. Case studies in reproducible research

The term "case studies" is used in a general sense to describe any study of reproducibility.[5] A reproduction is an attempt to arrive at comparable results with identical data using computational methods described in a paper. A refactor involves refactoring existing code into frameworks and reproducible best practices while preserving the original data. A replication involves generating new data and applying existing methods to achieve comparable results. A test of robustness applies various protocols, workflows, statistical models, or parameters to a given dataset to study their effect on results. A census is a high-level tabulation conducted by a third party. A survey is a questionnaire sent to practitioners. A case narrative is an in-depth first-person account. An independent discussion uses a secondary independent author to interpret the results of a study as a means to improve inferential reproducibility.

Big data, big science, and open data

The inability of third parties to reproduce results is not new to science,[21] but the scale of scientific endeavor and the level of data and method reuse suggest replication failures may damage the sustainability of certain disciplines, hence the term "reproducibility crisis." The problem of irreproducibility is compounded by the rise of "big data," in which very large, new, and often unique, disparate, or unformatted sources of data have been made accessible for analysis by third parties, and "big science," in which terabyte-scale datasets are generated and analyzed by multi-institutional collaborative research projects on specialized and possibly unique infrastructure. Metadata aspects of big data have been quantitatively studied concerning reuse,[22,23] but not reproducibility, despite some evidence that big data may play a role in spurious results associated with reporting bias.[24] Big data and big science have increased the demand for high-performance computing, specialized tools, and complex statistics, with attention to the growing popularity and application of machine learning and deep learning (ML/DL) techniques to these data sources. Such techniques typically train models on specific data subsets, and the models, as the end product of these methods, are often "black boxes," i.e., their internal predictors are not explainable (unlike older techniques such as regression) though they provide a good fit for the test data. Properly evaluating and reproducing studies that rely on such algorithms presents new challenges not previously encountered with inferential statistics.[25,26] Computational reproducibility is typically focused on the last analytic steps of what is often a labor-intensive scientific process that often originates from wet-lab protocols, fieldwork, or instrumentation and these last *in silico* steps present some of the more difficult problems both from technical and behavioral standpoints, because of the amount of entropy introduced by the sheer number of decisions made by an analyst. Developing solutions to make ML/DL workflows transparent, interpretable, and explorable to outsiders, such as peer reviewers, is an active area of research.[27]

The ability of third parties to reproduce studies relies on access to the raw data and methods employed by authors. Much to the exasperation of scientists, statisticians, and scientific software developers, the rise of "open data" has not been matched by "open analysis," as evidenced by several case studies.[20,28–30]

Missing data and code can obstruct the peer-review process, where proper review requires the authors to put forth the effort necessary to share a reproducible analysis. Software development practices, such as documentation and testing, are not a standard requirement of the doctoral curriculum, the peer-review process, or the funding structure, and as a result, the scientific community suffers from diminished reuse and reproducibility.[31] Sandve et al.[32] identified the most common sources of these oversights in "Ten Simple Rules for Reproducible Computational Research": lack of workflow frameworks, missing platform and software dependencies, manual data manipulation or forays into web-based steps, lack of versioning, lack of intermediates and plot data, and lack of literate programming or context can derail a reproducible analysis.

An issue distinct from the availability of source code and raw data is the lack of metadata to support reproducible research. We have observed that many of the findings from case studies in reproducibility point to missing methods details in an analysis, which can include software-specific elements such as software versions and parameters,[33] but also steps along the entire scientific process, including data collection and selection strategies, data processing provenance including hardware and statistical methods, and linking these elements to publication. We find the key concept connecting all of these issues is metadata.

An ensemble of dependency management and containerization tools already exist to accomplish narrow-sense reproducibility[34]: the ability to execute a packaged analysis with little effort from a third party. But context to allow for robustness and replicability, "broad-sense reproducibility," is limited without endorsement and integration of necessary metadata standards that support discovery, execution, and evaluation. Despite the growing availability of open-source tools, training, and better executable notebooks, reproducibility is still challenging.[35] In the following sections, we address these issues, first defining metadata, defining an "analytic stack" to abstract the steps of an *in silico* analysis, and then identifying and categorizing standards both established and in development to foster reproducibility.

Metadata

Over the past 25 years, metadata have gained acceptance as a key component of research infrastructure design. This trend is

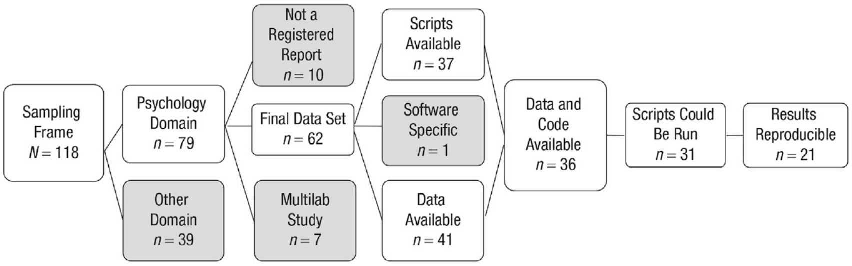


Figure 3. Reproducibility
Censuses like this one by Obels et al. measure data and code availability and reproducibility in this case over a corpus of 118 studies, 62 of which were psychology studies that had preregistered a Registered Report (RR).[20]

defined by numerous initiatives supporting the development and sustainability of hundreds of metadata standards, each with varying characteristics.[36,37] Across these developments, there is a general high-level consensus regarding the following three types of metadata standards[38,39]:

1. Descriptive metadata, supporting the discovery and general assessment of a resource (e.g., the format, content, and creator of the resource).
2. Administrative metadata, supporting technical and other operational aspects affiliated with resource use. Administrative metadata include technical, preservation, and rights metadata.
3. Structural metadata, supporting the linking among the components of a resource, so it can be fully understood.

There is also general agreement that metadata are a key aspect in supporting FAIR, as demonstrated by the FAIRsharing project (https://fairsharing.org), which divides standards types into "reporting standards" (checklists or templates, e.g., MIAME),[40] "terminology artifacts or semantics" (formal taxonomies or ontologies to disambiguate concepts, e.g., Gene Ontology),[41] "models and formats" (e.g., FASTA),[42] "metrics" (e.g., FAIRMetrics)[43] and "identifier schemata" (e.g., DOI)[44,45] (see Table 1).

Metadata are by definition structured. However, structured intermediates and results that are used as part of scientific analyses and employ encoding languages such as JSON or XML are recognized as primary data, not metadata. While an exhaustive distinction is beyond the scope of this paper, we define reproducible computational research metadata broadly as any structured data that aids reproducibility and that can conform to a standard. While this definition may seem liberal, we contend that metadata are the "glue" of reproducibility, and best identified by its function rather than its origins. This general understanding of metadata as a necessary component for research and data management and growing interest in reproducible computational research, together with the fact that there are few studies targeting metadata about the analytic stack that motivated the research presented in this paper.

Goals and methods

Our overall goal of this work is to review existing metadata standards and new developments that are directly applicable to reproducible computational research, identify gaps, discuss common threads among these efforts, and recommend next steps toward building a more robust infrastructure.

Our method is framed as a state-of-the-art review based on literature and ongoing software development in the scientific community. Review steps included: (1) defining key components of the analytic stack, and functions that metadata can support; (2) selecting exemplary metadata standards that address aspects of the identified functions; (3) assessing the applicability of these standards for supporting computational reproducibility functions; and (4) designing the corresponding metadata hierarchy. Our approach was informed, in part, by the Qin LIGO case study,[46] catalogs of metadata standards such as FAIRSharing, and comprehensive projects to bind semantic science such as Research Objects.[47] Compilation of core materials was accomplished mainly through literature searches but also perusal of code repositories, ontology catalogs, presentations, and Twitter posts. A "word cloud" of the most used abstract terms in the cited papers revealing most general terms is available in the code repository.

The RCR metadata stack

To define the key aspects of reproducible computational research, we have found it useful to break down the typical scientific computational analysis workflow, or "analytic stack," into five levels: (1) input, (2) tools, (3) reports, (4) pipelines, and (5) publications. These levels correspond loosely to the CRISP-DM data science process model (understanding, prep, modeling, evaluation, deployment),[48] scientific method (formulation, hypothesis, prediction, testing, analysis), and various research lifecycles as proposed by data curation communities (data search, data management, collection, description, analysis, archival, and publication)[49] and software development communities (plan, collect, quality control, document, preserve, use). However, unlike the steps in the life cycle, we do not emphasize a strong temporal order to these layers, but instead consider them simply interactive components of any scientific output.

RESULTS

In the course of our research, we found most standards, projects, and organizations were intended to address reproducibility issues that corresponded to specific activities in the analytic stack. However, metadata standards were unevenly distributed among the levels. Standards that could arguably be classified or repurposed into two or more areas were placed closest to their original intent. While we present the standards as a linear list of elements for the sake of clarity and comprehensibility, it is impossible to ignore their strongly intertwined nature. Pipelines, for example, also include data and code, journal articles, especially executable papers, and encompass metadata standards across many components. If communities are to embrace the RCR model, agreement is needed not just for individual metadata standards but also for elements that are used in concert.

Table 1. Types of FAIRsharing data and metadata standards

Type of standard	Purpose
Reporting standards	Ensure adequate metadata for reproduction
Terminology artifacts or semantics	Concept disambiguation and semantic relationships
Models and formats	Interoperability
Identifier schemata	Discovery

The synthesis below first presents a summary table (Table 2), followed by a more detailed description of each of the five levels, specific examples, and a forecast of future directions.

Input

Input refers to raw data from wet lab, field, instrumentation, or public repositories; intermediate processed files; and results from manuscripts. Compared with other layers of the analytic stack, input data garner the majority of metadata standards. Descriptive standards (metadata) enable the documentation, discoverability, and interoperability of scientific research and make it possible to execute and repeat experiments. Descriptive metadata, along with provenance metadata, also provides context and history regarding the source, authenticity, and life cycle of the raw data. These basic standards are usually embodied in the scientific output of tables, lists, and trees, which take form in files of innumerable file and database formats as input to reproducible computational analyses, filtering down to visualizations and statistics in published journal articles. Most instrumentation, field measurements, and wet lab protocols can be supported by metadata used for detecting anomalies such as batch effects and sample mix-ups.

Input metadata also serves to characterize gestalt aspects of datasets that may explain failures to replicate, such as a lack of population diversity in genomic studies,[91] or those that can quickly inform peer reviewers whether appropriate methods were employed for an analysis.

While metadata are often recorded from firsthand knowledge of the technician performing an experiment or the operator of an instrument, many forms of input metadata are in fact metrics that can be derived from the underlying data. This fact does not undermine the value of "derivable" metadata in terms of its importance for discovery, evaluation, and reproducibility.

Formal semantic ontologies represent one facet of metadata. The OBO Foundry[92] and NCBI BioPortal serve as catalogs of life science ontologies. The usage of these ontologies appears to follow a steep Pareto distribution, with the most popular ontologies generating thousands of citations, whereas the vast majority of NCBO's 883 ontologies have never been cited or mentioned.

Examples

In addition to being the oldest, and arguably most visible of reproducibility metadata standards, input metadata standards serve as a watershed for downstream reproducibility. To understand what input means for computational reproducibility, we examine three well-established examples of metadata standards from different scientific fields. Considering each of these standards reflects different goals and practical constraints of their respective fields, their longevity merits investigating what characteristics they have in common.

DICOM: An embedded file header. Digital Imaging and Communications in Medicine (DICOM) is a medical imaging standard introduced in 1985.[93] DICOM images require extensive technical metadata to support image rendering, and descriptive metadata to support clinical and research needs. These metadata coexist in the DICOM file header, which uses a group/element namespace to designate public restricted standard DICOM tags from private metadata. Extensive standardization of data types, called value representations (VRs) in DICOM, also follow this public/private scheme.[94] The public tags, standardized by the National Electrical Manufacturers Association (NEMA), have served the technical needs of both 2- and 3-dimensional images, as well as multiple frames, and multiple associated DICOM files or "series." Conversely, descriptive metadata have suffered from "tag entropy" in the form of missing, incorrectly filled, nonstandard, or misused tags by technicians manually entering in metadata.[95] This can pose problems both for clinical workflows as well as efforts to aggregate imaging data for data mining and machine learning. Advanced annotations supporting image segmentation and quantitative analysis have to conform to data structures imposed by the DICOM header format. This has made it necessary for programs such as 3DSlicer[96] and its associated plugins, such as dcqmi,[97] to develop solutions such as serializations to accommodate complex or hierarchical metadata.

EML: Flexible user-centric data documentation. Ecological Metadata Language (EML) is a common language for sharing ecological data.[50] EML was developed in 1997 by the ecology research community and is used for describing data in notable databases, such as the Knowledge Network for Biocomplexity (KNB) repository (https://knb.ecoinformatics.org/) and the Long Term Ecological Network (https://lternet.edu/). The standard enables documentation of important information about who collected the research data, when, and how, describing the methodology down to specific details and providing detailed taxonomic information about the scientific specimen being studied (Figure 4).

MIAME: A submission-centric minimal standard. Minimum Information About a Microarray Experiment (MIAME)[40] is a set of guidelines developed by the Microarray Gene Expression Data (MGED). society that has been adopted by many journals to support an independent evaluation of results. Introduced in 2001, MIAME allows public access to crucial metadata supporting gene expression data, i.e., quantitative measures of RNA transcripts via the Gene Expression Omnibus (GEO) database at the National Center for Biotechnology Information and European Bioinformatics Institute (EBI) ArrayExpress. The standard allows microarray experiments encoded in this format to be reanalyzed, supporting a fundamental goal of computational reproducibility: to support structured and computable experimental features.[99]

MIAME (Box 1) has been a boon to the practice of meta-analyses and harmonization of microarrays, offering essential array probeset, normalization, and sample metadata that make the over 2 million samples in GEO meaningful and reusable.[100] However, it should be noted that among MIAME and other Investigation/Study/Assay (ISA) standards that have followed suit,[101] none offer a controlled vocabulary for describing downstream computational workflows aside from slots to name the

Table 2. High-level summary

Metadata level	Description	Examples of metacontent	Examples of standards	Projects and organizations
1. Input	metadata related to raw data and intermediates	sequencing parameters, instrumentation, spatiotemporal extent	MIAME,* EML,* DICOM*, GBIF CIF ThermoML, CellML, DATS, FAANG, ISO/TC 276, NetCDF, OGC, GO	OBO, NCBO, FAIRsharing, Allotrope
2.Tools	metadata related to executable and script tools	version, dependencies, license, scientific domain	CRAN DESCRIPTION file,* Conda* meta.yaml/environment.yml, pip requirements.txt,* pipenv Pipfile/Pipfile.lock, Poetry pyproject.toml/poetry.lock, EDAM,* CodeMeta,* Biotoolsxsd, DOAP, ontosoft, SWO	Dockstore, Biocontainers
3.Statistical reports and notebooks	literate statistical analysis documents in Jupyter or knitr, overall statistical approach or rationale	session variables, ML parameters, inline statistical concepts	OBCS, STATO* SDMX DDI, MEX,* MLSchema, MLFlow,* Rmd YAML*	Neural Information Processing Systems Foundation
4.Pipelines, preservation, and binding	dependencies and deliverables of the pipeline, provenance	file intermediates, tool versions, deliverables	CWL,* CWLProv,* RO-Crate,* RO, WICUS, OPM, PROV-O, ReproZip Config, ProvOne, WES, BagIt, BCO, ERC	GA4GH, ResearchObjects, WholeTale, ReproZip
5.Publication	research domain, keywords, attribution	bibliographic, scientific field, scientific approach (e.g., "GWAS")	BEL,* Dublin Core, JATS, ONIX, MeSH, LCSH, MP, Open PHACTS, SWAN, SPAR, PWO, PAV	NeuroLibre, JOSS, ReScience, Manubot

Metadata standards, including MIAME,[40] EML,[50] DICOM,[51] GBIF,[52] CIF,[53] ThermoML,[54] CellML,[55] DATS,[56] FAANG,[57] ISO/TC 276,[58] GO,[41] Biotoolsxsd,[59] meta.yaml,[60] DOAP,[61] ontosoft,[62] EDAM,[63] SWO,[64] OBCS,[65] STATO,[66] SDMX,[67] DDI),[68] MEX,[69] MLSchema,[70] CWL,[71] WICUS,[72] OPM,[73] PROV-O,[74] CWLProv,[75] ProvOne,[76] PAV,[77] BagIt,[78] RO,[47] RO-Crate (abstract by Sefton et al., 2019), BCO,[79] Dublin Core,[80] JATS,[81] ONIX,[82] MeSH,[83] LCSH,[84] MP,[85] Open PHACTS,[86] BEL,[87] SWAN,[88] SPAR,[89] PWO.[90] *Standards that are featured within this article. Examples of all standards can be found at https://github.com/leipzig/metadata-in-rcr.

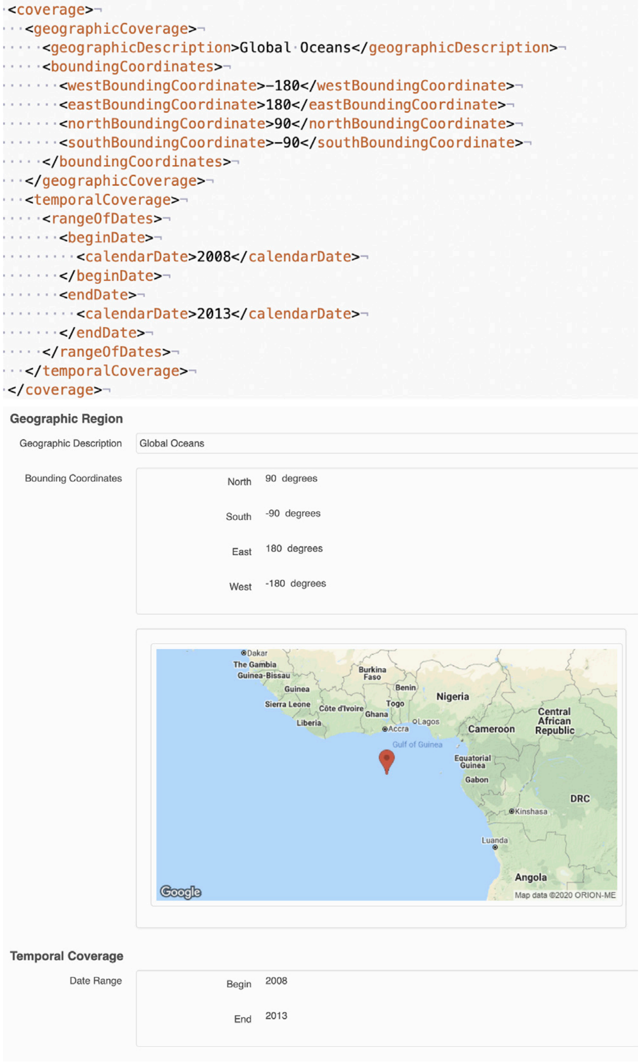
normalization procedure applied to what are essentially unitless intensity values.

Future directions: Encoding, findability, granularity, dimensionality

Metadata for input is developing along descriptive, administrative, and structural axes. Scientific computing has continuously and selectively adopted technologies and standards developed for the larger technology sector. Perhaps most salient from a development standpoint is the shift from extensible markup language (XML) to more succinct Javascript Object Notation (JSON) and Yet Another Markup Language (YAML) as preferred formats, along with requisite validation schema standards.[102]

The term "semantic web" describes an early vision of the Internet based on machine-readable contextual markup and semantically linked data using Uniform Resource Identifier (URI).[103] Schema.org, a consortium of e-commerce companies developing tags for markup and discovery, such as those recognized by Google Dataset Search,[104] has coalesced a stable set of tags that is expanding into scientific domains, demonstrating the potential for findability. Schema.org can be used to identify and distinguish inputs and outputs of analyses in a disambiguated and machine-readable fashion. DATS,[56] a Schema.org-compatible tag-suite describes fundamental metadata for datasets akin to that used for journal articles, especially to enable access to sensitive data. Combined with solutions for securely accessing analysis tools,[105,106] DATS can solve an often invoked impediment to reproducibility: that of unshareable data. The Open Research Knowledge Graph[107] (ORKG) aims to bring meaningfulness to scholarly documents in the same way as the semantic web for online documents. ORKG's structured semantic metadata on research contributions could not only improve findability and make scientific knowledge machine readable, but also mitigate reproducibility challenges.

Of increasing interest to the life sciences is the representation of phenotypic data to accompany various omics studies, as primary variables for genotype-by-environment studies, to control for possible confounds and random effects, and as labels for machine learning efforts toward genotype-to-phenotype prediction. Phenotypic metadata for human studies, ranging from basic demographics (e.g., sex, age) to complex attributes, such as disease, is often crucial to interpreting and reusing omics data. However, a study of 29 transcriptomics-based sepsis studies revealed 35% of the phenotypic information was lost in public repositories relative to their respective publication.[108] Efforts to standardize phenotypic information for plants, such as Minimal Information About Plant Phenotyping Experiment (MIAPPE), are challenged by a highly heterogeneous landscape of species, data types, and experimental designs.[109] This has required the development of the Plant Phenotyping Experiment Ontology (PPEO) data model with elements unique to botany.



```
·<coverage>¬
···<geographicCoverage>¬
·····<geographicDescription>Global·Oceans</geographicDescription>¬
·····<boundingCoordinates>¬
·······<westBoundingCoordinate>-180</westBoundingCoordinate>¬
·······<eastBoundingCoordinate>180</eastBoundingCoordinate>¬
·······<northBoundingCoordinate>90</northBoundingCoordinate>¬
·······<southBoundingCoordinate>-90</southBoundingCoordinate>¬
·····</boundingCoordinates>¬
···</geographicCoverage>¬
···<temporalCoverage>¬
·····<rangeOfDates>¬
·······<beginDate>¬
·········<calendarDate>2008</calendarDate>¬
·······</beginDate>¬
·······<endDate>¬
·········<calendarDate>2013</calendarDate>¬
·······</endDate>¬
·····</rangeOfDates>¬
···</temporalCoverage>¬
·</coverage>¬
```

Geographic Region

Geographic Description: Global Oceans

Bounding Coordinates

North 90 degrees

South -90 degrees

East 180 degrees

West -180 degrees

Temporal Coverage

Date Range

Begin 2008

End 2013

Figure 4. Ecological metadata language

Geographic and temporal EML metadata and the associated display on Knowledge Network for Biocomplexity (KNB) from Halpern et al.[98]

Finally, the growing scope for input metadata describing and defining unambiguous lab operations and protocols is important for reproducibility. One example of such an input metadata framework is the Allotrope Data Format, an HDF5 data structure, and accompanying ontology for chemistry protocols used in the pharmaceutical industry.[110] Allotrope uses the W3C Shapes Constraint Language (SHACL) to describe which RDF relationships are valid to describe lab operations.

Tools

Tool metadata refers to administrative metadata associated with computing environments, compiled executable software, and source code. In scientific workflows, executable and script-based tools are typically used to transform raw data into intermediates that can be analyzed by statistical packages and visualized as, e.g., plots or maps. Scientific software is written for a variety of platforms and operating systems; although Unix/Linux-based software is especially common, it is by no means a homogeneous landscape. In terms of reproducing and replicating studies, the specification of tools, tool versions, and parameters is paramount. In terms of tests of robustness (same data/different tools) and generalizations (new data/different tools), communicating the function and intent of a tool choice is also important and presents opportunities for metadata. Scientific software is scattered across many repositories in both source and compiled forms. Consistently specifying the location of software using URLs is neither trivial nor sustainable. To this end, a Software Discovery Index was proposed as part of the NIH Big Data To Knowledge (B2DK) initiative.[1] Subsequent work in the area cited the need for unique identifiers, supported by journals, and backed by extensive metadata.[111]

Examples

The landscape of metadata standards in tools is best organized into efforts to describe tools, dependencies, and containers.

CRAN, EDAM, and CodeMeta: Tool description and citation. Source code spans both tools and literate statistical reports, although for convenience we classify code as a subcategory of tools. Metadata standards do not exist for loose code, but packaging manifests with excellent metadata standards exist for several languages, such as R's Comprehensive R Archive Network (CRAN) DESCRIPTION files (Box 2).

Recent developments in tools metadata have focused on tool description, citation, dependency management, and containerization. The last two advances, exemplified by the Conda and Docker projects (described below), have largely made computational reproducibility possible, at least in the narrow sense of being able to reliably version and install software and related dependencies on other people's machines. Often small changes in software and reference data can have substantial effects on an analysis.[113] Tools like Docker and Conda respectively make the computing environment and version pinning software tenable, thereby producing portable and stable environments for reproducible computational research.

The EMBRACE Data And Methods (EDAM) ontology provides high-level descriptions of tools, processes, and biological file formats.[63] It has been used extensively in tool recommenders,[114] tool registries,[115] and within pipeline frameworks and workflow languages.[116,117] In the context of workflows, certain tool combinations tend to be chained in predictable usage patterns driven by application; these patterns can be mined for tool recommender software used in workbenches.[118]

CodeMeta[119] prescribes JSON-LD (JSON for Linked Data) standards for code metadata markup. While CodeMeta is not itself an ontology, it leverages Schema.org ontologies to provide language-agnostic means of describing software as well as "crosswalks" to translate manifests from various software repositories, registries, and archives into CodeMeta (Box 3).

Considerable strides have been made in improving software citation standards,[121] which should improve the provenance of methods sections that cite those tools that do not already have accompanying manuscripts. Code attribution is implicitly fostered by the application of large-scale data mining of code repositories, such as Github is the generation of dependency networks,[122] measures of impact,[123] and reproducibility censuses.[124]

Dependency and package management metadata. Compiled software often depends on libraries that are shared by many

Box 1. An example of MIAME in MINiML formathttps://www.ncbi.nlm.nih.gov/geo/info/MINiML_Affy_example.txt

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<MINiML
  xmlns="https://www.ncbi.nlm.nih.gov/geo/info/MINiML"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://www.ncbi.nlm.nih.gov/geo/info/MINiML
https://www.ncbi.nlm.nih.gov/geo/info/MINiML.xsd"
  version="0.5.0" >
  <Contributor iid="contrib1">
    <Person><First>Jun</First><Last>Shima</Last></Person>
  </Contributor>
  <Contributor iid="contrib2">
    <Person><First>Fumiko</First><Last>Tanaka</Last></Person>
  </Contributor>
  <Contributor iid="contrib3">
    <Person><First>Akira</First><Last>Ando</Last></Person>
  </Contributor>
  <Contributor iid="contrib4">
    <Person><First>Toshihide</First><Last>Nakamura</Last></Person>
  </Contributor>
  <Contributor iid="contrib5">
    <Person><First>Hiroshi</First><Last>Takagi</Last></Person>
  </Contributor>
  <Database iid="GEO">
    <Name>Gene Expression Omnibus (GEO)</Name>
    <Public-ID>GEO</Public-ID>
    <Organization>NCBI NLM NIH</Organization>
    <Web-Link>https://www.ncbi.nlm.nih.gov/geo</Web-Link>
    <Email>geo@ncbi.nlm.nih.gov</Email>
  </Database>
  <Platform iid="GPL90">
    <Accession database="GEO">GPL90</Accession>
  </Platform>
  <Sample iid="Sample1">
    <Title>
before fermentation
    </Title>
    <Channel-Count>1</Channel-Count>
    <Channel position="1">
      <Source>mRNA T128</Source>
      <Organism>Saccharomyces cerevisiae</Organism>
      <Characteristics>
```

programs on an operating system. Conflicts between versions of these libraries, and software that demands obscure or outdated versions of these libraries, are a common source of frustration for users who install scientific software and a major hurdle to distributing reproducible code. Until recently, installation woes and "dependency hell" were considered a primary stumbling block to reproducible research.[125] Software written in high-level languages such as Python and R has traditionally relied on language-specific package management systems and repositories, e.g., pip and PyPI for Python, and the install.packages() function and CRAN for R. The complexity yet unavoidability of controlling dependencies led to competing and evolving tools, such as pip, Pipenv, and Poetry in the Python community, and even different conceptual approaches, such as the CRAN time machine. In recent years, a growing number of scientific software projects use combinations of Python and compiled software. The Conda project (https://conda.io) was developed to provide a universal solution for compiled executables and script dependencies written in any language. The elegance of providing a single requirements file has contributed to Conda's rapid adoption for domain-specific library collections such as Bioconda,[126] which are maintained in "channels" that can be subscribed and prioritized by users.

Fledgling standards for containers. For software that requires a particular environment and dependencies that may conflict with an existing setup, a lightweight containerization layer provides a means of isolating processes from the underlying operating system, basically providing each program with its own miniature

Box 2. R description An R package DESCRIPTION file from DESeq2.[112]

```
Package: DESeq2
Type: Package
Title: Differential gene expression analysis based on the negative
  binomial distribution
Version: 1.33.1
Authors@R: c(
  person("Michael", "Love", email="michaelisaiahlove@gmail.com", role =
c("aut","cre")),
  person("Constantin", "Ahlmann-Eltze", role = c("ctb")),
  person("Kwame", "Forbes", role = c("ctb")),
  person("Simon", "Anders", role = c("aut","ctb")),
  person("Wolfgang", "Huber", role = c("aut","ctb")),
  person("RADIANT EU FP7", role="fnd"),
  person("NIH NHGRI", role="fnd"),
  person("CZI", role="fnd"))
Maintainer: Michael Love<michaelisaiahlove@gmail.com>
Description: Estimate variance-mean dependence in count data from
  high-throughput sequencing assays and test for differential
  expression based on a model using the negative binomial
  distribution.
License: LGPL (>= 3)
VignetteBuilder: knitr, rmarkdown
Imports: BiocGenerics (>= 0.7.5), Biobase, BiocParallel, genefilter,
  methods, stats4, locfit, geneplotter, ggplot2, Rcpp (>= 0.11.0)
Depends: S4Vectors (>= 0.23.18), IRanges, GenomicRanges,
  SummarizedExperiment (>= 1.1.6)
Suggests: testthat, knitr, rmarkdown, vsn, pheatmap, RColorBrewer,
  apeglm, ashr, tximport, tximeta, tximportData, readr, pbapply,
  airway, pasilla (>= 0.2.10), glmGamPoi, BiocManager
LinkingTo: Rcpp, RcppArmadillo
URL: https://github.com/mikelove/DESeq2
biocViews: Sequencing, RNASeq, ChIPSeq, GeneExpression, Transcription,
  Normalization, DifferentialExpression, Bayesian, Regression,
  PrincipalComponent, Clustering, ImmunoOncology
RoxygenNote: 7.1.1
Encoding: UTF-8
```

operating system. The ENCODE project[127] provided a virtual machine for a reproducible analysis that produced many figures featured in the article and serves as one of the earliest examples of an embedded virtual environment. While originally designed for deploying and testing e-commerce web applications, the Docker containerization system has become useful for scientific environments where dependencies and permissions become unruly. Several papers have demonstrated the usefulness of Docker for reproducible workflows[125,128] and as a central unit of tool distribution.[129,130]

Conda programs can be trivially Dockerized, and every BioConda package gets a corresponding BioContainer[131] image built for Docker and Singularity, a similar container solution designed for research environments. Because Dockerfiles are similar to shell scripts, Docker metadata are an underutilized resource and one that may need to be further leveraged for reproducibility. Docker does allow for arbitrary custom key-value metadata (labels) to be embedded in containers (Box 4). The Open Container Initiative's Image Format Specification (https://github.com/opencontainers/image-spec/) defines pre-defined keys, e.g., for authorship, links, and licenses. In practice, the now deprecated Label Schema (http://label-schema.org/rc1/) labels are still pervasive, and users may add arbitrary labels with prepended namespaces. It should be noted that containerization is not a panacea and Dockerfiles can introduce irreproducibility and decay if contained software is not sufficiently pinned (e.g., by using so-called lockfiles) and installed from sources that are available in the future.

Future directions

Automated repository metadata. Source code repositories such as Github and Bitbucket are designed for collaborative development, version control, and distribution and as such do not enforce any reproducible research standards that would be useful for evaluating scientific code submissions. As a corresponding example to the NLP above, there are now efforts to mine source code repositories for discovery and reuse.[132]

Data as a dependency. "Data libraries," which pair data sources with common programmatic methods for querying them, are

Box 3. CodeMeta A snippet of CodeMeta JSON file from Price et al.[120] using Schema.org contextual tags.

```
{
  "@context": [
    "https://doi.org/10.5063/schema/codemeta-2.0",
    "http://schema.org"
],
  "@type": "SoftwareSourceCode",
  "identifier": "baydem",
  "description": "Bayesian tools for reconstructing past and present\n
demography.",
  "name": "baydem: Bayesian Tools for Reconstructing Past and Present Demography",
  "license": "https://spdx.org/licenses/MIT",
  "version": "0.1.0",
  "programmingLanguage": {
    "@type": "ComputerLanguage",
    "name": "R",
    "url": "https://r-project.org"
},
"runtimePlatform": "R version 4.0.2 (2020-06-22)",
"author": [
  {
    "@type": "Person",
    "givenName": ["Michael", "Holton"],
    "familyName": "Price",
    "email": "michaelholtonprice@sgmail.com"
  }
]
```

very popular in centralized open source repositories such as Bioconductor,[133] and scikit-learn,[134] despite often being large downloads. Tierney and Ram provide a best practices guide to the organization and necessary metadata for data libraries and independent datasets.[135] Ideally, users and data providers should be able to distribute data recipes in a decentralized fashion, for instance, by broadcasting data libraries in user channels. Most raw data include a limited number of formats, but ideally, data should be distributed in packages bound to a variety of tested formatters. One solution, Gogetdata[136] is a project that can be used to specify versioned *data* prerequisites to coexist with software within the Conda requirements specification file. A private company called Quilt is developing similar data-as-a-dependency solutions bound to a cloud computing model. A similar effort, Frictionless Data, focuses on JSON-encoded schemas for tabular data and data packages featuring a manifest to describe constitutive elements. From a Docker-centric perspective, the Open Container Initiative[137] is working to standardize "filesystem bundles": the collection of files in a container and their metadata. In particular, container metadata are critical for relating the contents of a container to its source code and version, its relationship with other containers, and how to use the container. Ongoing research on container preservation[138,139] can introduce new structural metadata on container usage to avoid container images becoming just a binary bitstream when they are archived, but instead they remain understandable and even actionable.

Neither Conda nor Docker is explicitly designed to describe software with fixed metadata standards or controlled vocabularies. This suggests that a centralized database should serve as a primary metadata repository for tool information, rather than a source code repository, package manager, or

Box 4. Excerpt from a Dockerfile: LABEL instruction with image metadata Source: https://github.com/nuest/ten-simple-rules-dockerfiles/blob/master/examples/text-analysis-wordclouds_R-Binder/Dockerfile.

```
LABEL maintainer="daniel.nuest@uni-muenster.de" \
  Name="Reproducible research at GIScience - computing environment" \
  org.opencontainers.image.created="2020-04" \
  org.opencontainers.image.authors="Daniel Nüst" \
org.opencontainers.image.url="https://github.com/nuest/reproducible-research-at-giscience/blob/master/
Dockerfile" \
org.opencontainers.image.documentation="https://github.com/nuest/reproducible-research-at-giscience/" \
  org.opencontainers.image.licenses="Apache-2.0" \
  org.label-schema.description="Reproducible workflow image (license: Apache 2.0)"
```

container store. An example of such a database is the GA4GH Dockstore,[140] a hub and associated website that allows for a standardized means of describing and invoking Dockerized tools as well as sharing workflows based on them.

Statistical reports and notebooks

Statistical reports and notebooks serve as an annotated session of an analysis. Though they typically use input data that have been processed by scripts and workflows (see below), they can be characterized as a step in the workflow rather than apart from it, and for some smaller analyses, all processing can be done within these notebooks. Statistical reports and notebooks occupy an elevated reputation as being an exemplar of reproducible best practices, but they are not a reproducibility panacea and can introduce additional challenges, one reason being the metadata supporting them is surprisingly sparse.

Statistical reports that use ‘‘literate programming,’’ combining statistical code with descriptive text, markup, and visualizations, have been a standard for statistical communication since the advent of Sweave.[141] Sweave allowed R and LaTeX markup to be mixed in chunks, allowing the adjacent contextual descriptions of statistical code to serve as guideposts for anyone reading a Sweave report, typically rendered as PDF. An evolution of Sweave, knitr,[142] extended choices of both markup (allowing Markdown) and output (HTML) while enabling tighter integration with integrated development environments such as RStudio.[143] A related project that started in the Python ecosystem but now supports several kernels, Jupyter,[144] combined the concept of literate programming with an REPL (read-eval-print loop) in a web-based interactive session in which each block of code is kept stateful and can be reevaluated. These live documents are known as "notebooks." Notebooks provide a means of allowing users to directly analyze data programmatically using common scripting languages, and access more advanced data science environments such as Spark, without requiring data downloads or localized tool installation if run on cloud infrastructures. Using preloaded libraries, cloud-based notebooks can alleviate time-consuming permissions recertification, downloading of data, and dependency resolution, while still allowing persistent analysis sessions. Dataset-specific Jupyter notebooks "spawned" for thousands of individuals temporarily have been enabled as companions for *Nature* articles[145] and are commonly used in education. Cloud-based notebooks have not yet been extensively used in data portals, but they represent the analytical keystone to the decade-long goal of "bringing the tools to the data." Notebooks offer possibilities over siloed installations in terms of eliminating the data science bottlenecks common to data analyses: cloud-based analytic stacks, cookbooks, and shared notebooks.

Collaborative notebook sharing has been used to accelerate the analysis cycle by allowing users to leverage existing code. The predictive analytics platform Kaggle employs an open implementation of this strategy to host data exploration events. This approach is especially useful for sharing data cleaning tasks—removing missing values, miscategorizations, and phenotypic standardization, which can represent 80% of effort in an analysis.[146] Sharing capabilities in existing open-source notebook platforms are at a nascent stage, but this presents possibilities for reproducible research environments to flourish. One promising project in this area is Binder, which allows users to instantiate live Jupyter notebooks and associated Dockerfiles stored on Github within a Kubernetes-backed service.[147],[148]

At face value, reports and notebooks resemble source code or scripts, but as the vast majority of statistical analysis and machine learning education and research is conducted in notebooks, they represent an important area for reproducibility.

Examples

R Markdown headers. As we mentioned, statistical reports and notebooks do not typically leverage structured metadata for reproducibility. R Markdown–based reports, such as those processed by knitr, do have a YAML-based header or front matter (Box 5). These are used for a wide variety of technical parameters for controlling display options, for providing structured metadata on authors, e.g., when used for scientific publications with the *rticles* package,[149] or for parameterizing the included workflow (https://rmarkdown.rstudio.com/developer_parameterized_reports.html). However, no schema or standards exist for their validation beyond syntax, and different tools freely extend them for their own needs.

Statistical and machine learning metadata standards. The intense interest paired with the competitive nature of machine learning and deep learning conferences such as Neurips demands high reproducibility standards.[150] Given the predominance of notebooks for disseminating machine learning workflow, we focused our attention on finding statistical and machine learning metadata standards that would apply to content found with notebooks. The opacity, rapid proliferation, and multifaceted nature of machine learning and data mining statistical methods to nonexperts suggest it is necessary to begin cataloging and describing them at a more refined level than crude categories (e.g., clustering, classification, regression, dimension reduction, feature selection). So far, the closest attempt to decompose statistics in this manner is the STATO statistical ontology (http://stato-ontology.org/), which can be used to semantically, rather than programmatically or mathematically, define all aspects of a statistical model and its results, including assumptions, variables, covariates, and parameters (Figure 5). While STATO is currently focused on univariate statistics, it represents one possible conception for enabling broader reproducibility than simply relying on specific programmatic implementations of statistical routines.

MEX is designed as a vocabulary to describe the components of machine learning workflows. The MEX vocabulary builds on PROV-O to describe specific machine learning concepts such as hyperparameters and performance measures and includes a decorator class to work with Python.

Future directions: Parameter tracking

MLFlow[152] is designed specifically to handle hyperparameter tracking for machine learning iterations or "runs" performed in the Apache Spark, but also tracks arbitrary artifacts and metrics associated with these. The metadata format that MLFlow uses exposes variables that are explored and tuned by end-users (Box 6).

Pipelines

Most scientific analyses are conducted in the form of pipelines, in which a series of transformations is performed on raw data, followed by statistical tests and report generation. Pipelines are also referred to as "workflows," which sometimes also

Box 5. A YAML-based R Markdown header for demonstration purposes Full document shared in this papers’ repository at https://github.com/leipzig/metadata-in-rcr/.

```
— title: "A title for the analysis" # author metadata, esp. used for scientific articles
author:
  - name: Jeremy Leipzig
  footnote: Corresponding author
  affiliation: "Metadata Research Center, Drexel University, College of Computing and Informatics,
  orcid: "0000-0001-7224-9620"
  - name: Daniel Nüst
  affiliation: "Institute for Geoinformatics, University of Münster, Germany"
  orcid: "0000-0002-0024-5046"
  email:daniel.nuest@uni-muenster.de
# parameters to manipulate workflow; defaults can be changed when compiling the document
params:
  year: 2020
  region: "Europe"
  printcode: TRUE
  data: file.csv
  max_n: 42
# configuration and styling of different output document formats
output:
  html_document:
    theme: lumen
    toc: true
    toc_float:
      collapsed: false
    code_folding: show
    self_contained: true
  pdf_document:
    toc: yes
    fig_caption: yes
    df_print: kable
linkcolor: blue
# field values can be generated from code
date: "`r format(Sys.time(), ’%d %B, %Y’)`"
```

encompasses steps outside an automated computational process. Pipelines represent the computation component of many papers, in both basic research and tool papers. Pipeline frameworks or scientific workflow management systems (SWfMS) are platforms that enable the creation and deployment of reproducible pipelines in a variety of computational settings including cluster and cloud parallelization. The use of pipeline frameworks, as opposed to standalone scripts, has recently gained traction, largely due to the same factors (big data, big science) driving the interest of reproducible research. Although frameworks are not inherently more reproducible than shell scripts or other scripted ad hoc solutions, use of them tends to encourage parameterization and configuration that promote reproducibility and metadata. Pipeline frameworks are also attractive to scientific workflows in that they provide tools for the reentrancy—restarting a workflow where it left off, implicit dependency resolution—allowing the framework engine to automatically chain together a series of transformation tasks, or "rules," to produce a user-supplied file target. Collecting and analyzing provenance, which refers to the record of all activities that go into producing a data object, is a key challenge for the design of pipelines and pipeline frameworks.

The number and variety of pipeline frameworks have increased dramatically in recent years; each framework built with design philosophies that offer varying levels of convenience, user-friendliness, and performance. There are also tradeoffs between the dynamicity of a framework, in terms of its ability to behave flexibly (e.g., skip certain tasks, re-use results from a cache) based on input, that will affect the apparent reproducibility and the run-level metadata that is required to inspire confidence in an analyst’s ability to infer how a pipeline behaved in a particular situation. Leipzig[153] reviewed and categorized these frameworks into three key dimensions: using an implicit or explicit syntax; using a configuration, convention, or class-based design paradigm; and offering a command line or workbench interface.

“Convention-based frameworks” are typically implemented in a domain-specific language, a meaningful symbol set to represent rule input, output, and parameters that augment existing scripting languages to provide the glue to create workflows. These can often mix shell-executable commands with internal script logic in a flexible manner. “Class-based pipeline frameworks” augment programming languages to offer fine-granularity means of efficient distribution of data for high-performance

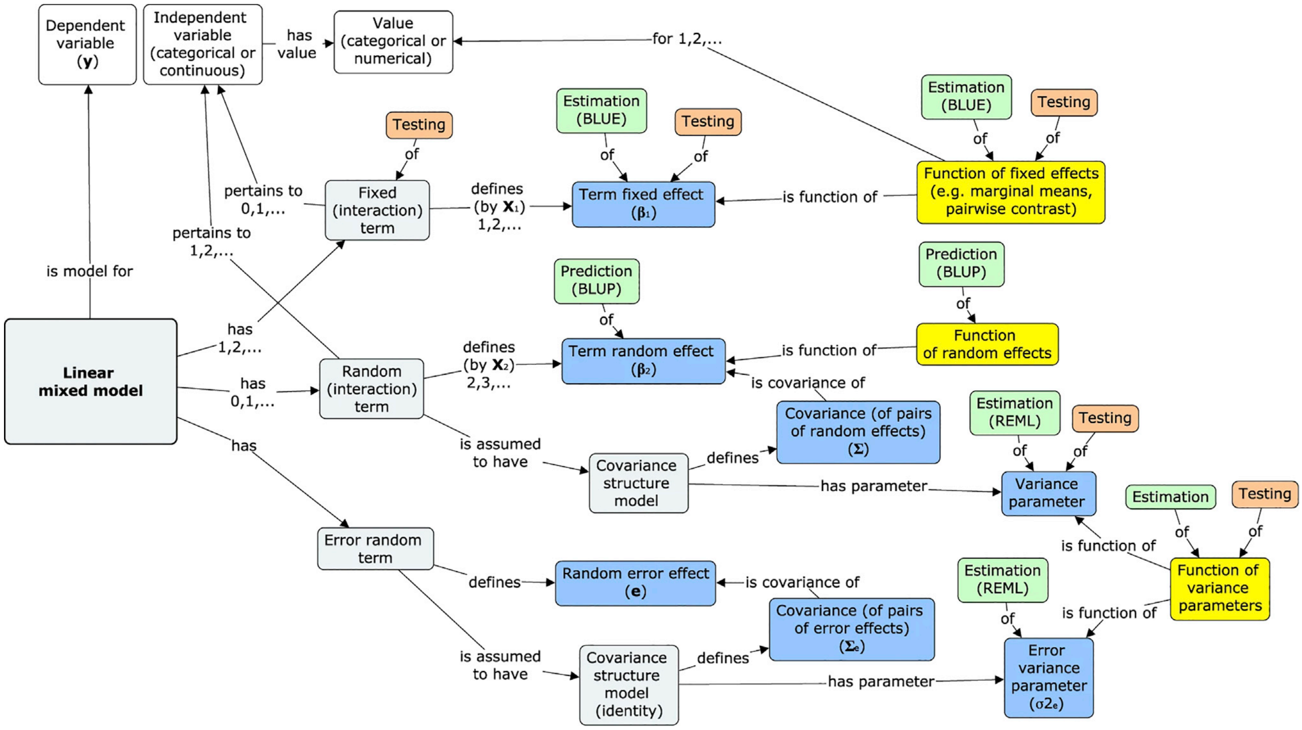


Figure 5. STATO
Concepts describing a linear mixed model used by STATO[151]

cluster computing frameworks such as Apache Spark. “Configuration-based frameworks” abstract pipelines into configuration files, typically XML or JSON which contain little or no code. Workbenches such as Galaxy,[154] Kepler,[155] KNIME,[156] Taverna,[157] and commercial workbenches, such as Seven Bridges Genomics and DNANexus, typically offer canvas-like graphical user interfaces by which tasks can be connected and always rely on configuration-based tool and workflow descriptors. Customized workbenches configured with a selection of pre-loaded tools and workflows and paired with a community web portal are often termed "science gateways."

Examples

CWL: A configuration-based framework for interoperability. The Common Workflow Language (CWL)[71] is a specification for tools and workflows to share across several pipeline frameworks, adopted by several workbenches. CWL manages the exacting specification of file inputs, outputs, parameters that are "operational metadata" used by the workflow machinery to communicate with the shell and executable software (Figure 6). While these metadata are primarily operational in nature and rarely accessed outside the context of a compatible runner such as Rabix[158] or Toil,[159] CWL also enables tool metadata in the form of versioning, citation, and vendor-specific fields that may differ between implementations.

Using this metadata, an important aspect of CWL is the focus on richly describing tool invocations both for reproducibility and documentation purposes, with tools referenced as retrievable Docker images or Conda packages, and identifiers to EDAM,[63] ELIXIR’s bio.tools[59] registry and Research Resource Identifiers (RRIDs).[161] This wrapping of command line tool interfaces is used by GA4GH Dockstore[140] for providing a uniform executable interface to a large variety of computational tools even outside workflows.

While there are many other configuration-based workflow languages, CWL is notable for the number of parsers that support its creation and interpretation, and an advanced linked data validation language, called Schema Salad. Together with supporting projects, such as Research Objects, the CWL appears amenable to being used as metadata.

Future directions

Interoperable script and workflow provenance. For future metadata to support pipeline reproducibility, it must accommodate a huge menagerie of solutions that coexist inside a number of computing environments. Large organizations have been encouraging the use of cloud-based data commons, but solutions that target the majority of published scientific analysis must address the fact that many if not most of them will not use a data commons or even a pipeline framework. Because truly reproducible research implies evaluation by third parties, portability is an ongoing concern.

Pimentel et al. reviewed and categorized 27 approaches to collecting provenance from scripts.[162] A wide variety of relational databases and proprietary file formats are used to store, distribute, visualize, version, and query provenance from these tools. The authors found that while four approaches—RDataTracker,[163] SPADE,[164] StarFlow,[165] and YesWorkflow[166]—natively adopt interoperable W3C PROV or OPM standards as export, most were designed for internal usage and did not enable sharing or comparisons of provenance. In part, these limitations

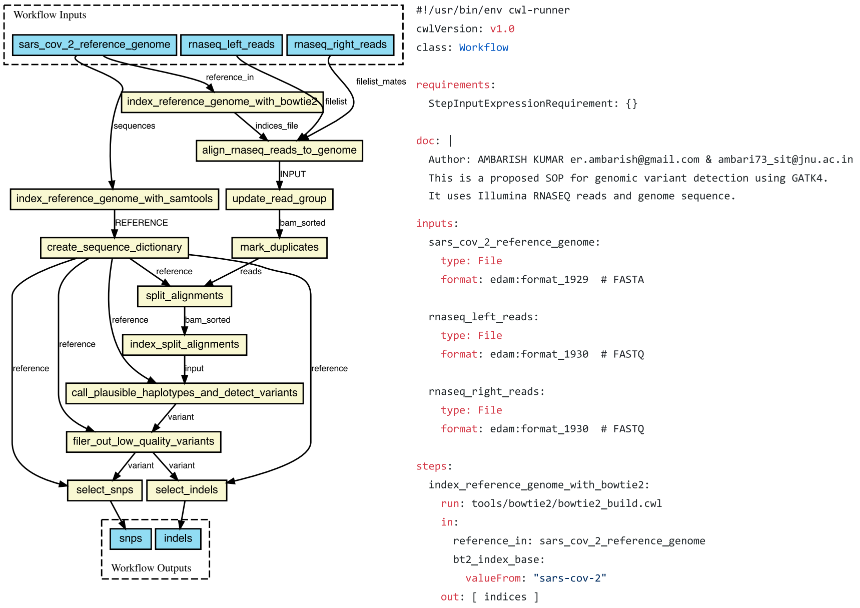
Box 6. MLflow snippet showing exposed hyperparameters

```
name: HyperparameterSearch
conda_env: conda.yaml
entry_points:
  # train Keras DL model
  train:
    parameters:
      training_data: {type: string, default: "./datasets/wine-quality.csv"}
      epochs: {type: int, default: 32}
      batch_size: {type: int, default: 16}
      learning_rate: {type: float, default: 1e-1}
      momentum: {type: float, default: .0}
      seed: {type: int, default: 97531}
    command: "python train.py {training_data}
          --batch-size {batch_size}
          --epochs {epochs}
          --learning-rate {learning_rate}
          --momentum {momentum}"
```

are related to primary goals and scope of these provenance tracking tools.

For analyses that use workflows, a prerequisite for reproducible research is the ability to reliably share "workflow enactments," or runs that encompass all elements of the analytic stack. Unlike pipeline frameworks geared toward cloud-enabled scalability, compatibility with executable command-line arguments and programmatic extensibility afforded by DSLs, Vistrails was designed explicitly to foster provenance tracking and querying, both prospective and retrospective.[167] As part of the WINGS project, Garijo et al.[168] use linked-data standards—OWL, PROV, and RDF—to create a framework-agnostic Open Provenance for Workflows (OPMW) for greater semantic possibilities for user needs in workflow discovery and publishing. The CWLProv[75] project implements a CWL-centric and RO-based solution with a goal of defining a format of implementing retrospective provenance.

Packaging and binding building blocks. While we have attempted to classify metadata across layers of the analytic stack, there are a number of efforts to tie or bind all these metadata that define a research compendia explicitly. A research compendium (RC) is a container for building blocks of a scientific workflow. Originally defined by Gentleman and Temple Lang as a means for distributing and managing documents, data, and computations using a programming language's packaging mechanism, the term is now used in different communities to provide code, data, and documentation (including scientific manuscripts) in a meaningful and useable way (https://research-compendium.science/). A best practice compendium includes environment configuration files (see above), has files that are under version



```
#!/usr/bin/env cwl-runner
cwlVersion: v1.0
class: Workflow

requirements:
  StepInputExpressionRequirement: {}

doc: |
  Author: AMBARISH KUMAR er.ambarish@gmail.com & ambari73_sit@jnu.ac.in
  This is a proposed SOP for genomic variant detection using GATK4.
  It uses Illumina RNASEQ reads and genome sequence.

inputs:
  sars_cov_2_reference_genome:
    type: File
    format: edam:format_1929  # FASTA

  rnaseq_left_reads:
    type: File
    format: edam:format_1930  # FASTQ

  rnaseq_right_reads:
    type: File
    format: edam:format_1930  # FASTQ

steps:
  index_reference_genome_with_bowtie2:
    run: tools/bowtie2/bowtie2_build.cwl
    in:
      reference_in: sars_cov_2_reference_genome
      bt2_index_base:
        valueFrom: "sars-cov-2"
    out: [ indices ]
```

Figure 6. Common workflow language
Snippets of a COVID-19 variant detection CWL workflow and the workflow as viewed through the cwl-viewer.[160]
Note the EDAM file definitions.

Box 7. erc.yml example fileSee the specification at https://o2r.info/erc-spec/.

```
id: b9b0099e-9f8d-4a33-8acf-cb0c062efaec
spec_version: 1
licenses:
  code: Apache-2.0
  data: data-licenses.txt
  text: "Creative Commons Attribution 2.0 Generic (CC BY 2.0)"
  metadata: "see metadata license headers"
```

control, and uses accessible plain text formats. Instead of a formal workflow specification, inputs, outputs, and control files and the required commands are documented for human users in a README file. While an RC can take many forms, the flexibility is also a challenge for extracting metadata. The Executable Research Compendium (ERC) formalizes the RC concept with an R Markdown notebook for the workflow and a Docker container for the runtime environment.[169] A YAML configuration file connects these parts (Box 7), configures the document to be displayed to a human user, and provides minimal metadata on licenses. The concept of bindings connects interactive parts of an ERC workflow with the underlying code and data.[170]

Instead of trying to establish a common standard and single point for metadata, the ERC intentionally skips formal metadata and exports the known information into multiple output files and formats, such as Zenodo metadata as JSON or Datacite as XML, accepting duplication for the chance to provide usable information in the long term.

Perhaps the most prominent realization of the RC concept is Research Objects[171] and the subsequent RO-Crate (Carragáin, E.Ó., et al., 2019, BOSC, abstract) projects (Box 8), which strive to be comprehensive solutions for binding code, data, workflows, and publications into a metadata-defined package. RO-Crate is lightweight JSON-LD (javascript object notation linked data) that supports Schema.org concepts to identify and describe all constituent files from the analytic stack and various people, publication, and licensing metadata, as well as provenance both between workflows and files and across crate versions.

An alternative approach to binding is to leverage existing work in "application profiles,"[172] a highly customizable means of combining namespaces from different metadata schemas. Application profiles follow along the Singapore Framework (Figure 7), and guidelines supported by the Dublin Core Metadata Initiative (DCMI).

Publication

Our conception of the analytic stack points to the manuscript as the final product of an analysis. Due to the requirements of cataloging, publishing, attribution, and bibliographic management, journals employ a robust set of standards including MARC21 and ISO_2709 for citations, and Journal Article Tag Suite (JATS) for manuscripts. Library science has been an early adopter of many metadata standards and encoding formats (e.g., XML) later used throughout the analytic stack. Supplementing and extending these standards to accommodate reproducible analyses connected or even embedded in publications is an open area for development.

For the purposes of reproducibility, we are most interested in finding publication metadata standards that attempt to support structured results as a "first-class citizen," essentially input metadata but for integration into the manuscript.

The methods section of a peer-reviewed article is the oldest and often the sole source of metadata related to an analysis. However, methods sections and other free-text supplementals are notoriously poor and unreliable examples of reproducible computational research, as evidenced by the Amgen findings and numerous reproduction studies. A number of text mining efforts have sought to extract details of the software used in analyses directly from methods sections for purposes of survey[173,174] and recommendation[175] using natural language processing (NLP). The ProvCaRe database and web application extend this to both computational and clinical findings by using a wide-ranging corpus of provenance terms and extending existing PROV-O ontology.[176] While these efforts are noble, they can never entirely bridge the gap between human-readable protocols and machine-readable metadata schemes.

Journals share an important responsibility to enforce and incentivize reproducible research,[177] but most peer-reviewed publications have been derelict in this role. While many have raised standards for open data access, "open analysis" is still an alien concept to many journals and code execution during peer review a rare understudied practice.[178] Some journals, such as *Nature Methods*, do require authors to submit source code.[179] Of the most prestigious life science journals (*Nature*, *Science*, *Cell*), the requirements vary considerably and it is not clear how these guidelines are actually enforced.[180] Few journals have clear reproduction policies.[181] The CODECHECK initiative aims to establish a minimum workflow for independent code execution during peer review to be adopted by publishers.[181] A YAML configuration file (https://codecheck.org.uk/spec/config/1.0/) in each code repository provides metadata for a check. The metadata connects the publication's and the reproduction certificate's DOIs, provides authorship information, lists the output files that were reproduced in a manifest, and is published in a register listing all checks (https://codecheck.org.uk/register/).

Container portals, package repositories, and workbenches do provide some additional inherent structure that would be useful for journals to require, but these often lack any binding with notebooks or elegant routes to report generation that would guarantee the scientific code matches the results contained with a manuscript. We should not underestimate the technical challenges of building and maintaining these advances. Computational provenance between all figures and tables in a manuscript and the underlying analysis is an open area of research that we discuss below.

Box 8. RO-Crate metadata

```
{ "@context": "https://w3id.org/ro/crate/1.0/context",
  "@graph": [
  {
    "@type": "CreativeWork",
    "@id": "ro-crate-metadata.jsonld",
    "conformsTo": {"@id": "https://w3id.org/ro/crate/1.0"},
    "about": {"@id": "./"}
  },
  {
    "@id": "./",
    "identifier": "https://doi.org/10.4225/59/59672c09f4a4b",
    "@type": "Dataset",
    "datePublished": "2020",
    "name": "Data files associated with the manuscript:The Role of Metadata in
  Reproducible Computational Research",
    "description": "Palliative care planning for nursing home residents with
  advanced dementia ...",
    "license": {"@id": "https://creativecommons.org/licenses/by-nc-sa/3.0/au/"},
    "hasPart": [
      {
          "@id": "src/"
      },
      {
          "@id": "metadata_examples/"
      }
    ]
  },
  {
    "@id": "https://creativecommons.org/licenses/by-nc-sa/4.0/au/",
    "@type": "CreativeWork",
    "description": "Creative Commons Attribution 4.0 International Public License
  (Public License)",
    "identifier": "https://creativecommons.org/licenses/by-nc-sa/3.0/au/",
    "name": "Attribution-NonCommercial-ShareAlike 3.0 Australia (CC BY-NC-SA 3.0 AU)"
    }
  ]
}
```

Examples

Formalization of the results of biological discovery. In the scientific literature, authors must not only outline the formulation of their experiments, their execution, and their results, but also an interpretation of the results with respect to an overarching scientific goal. Due to the lack of specificity of prose and the needless jargon endemic to modern scientific discourse, both the goals and interpretation of results are often obfuscated such that the reader must exert considerable effort to understand. This burden is further exacerbated by the acceleration of the growth of the body of scientific literature. As a result, it has become overwhelming, if not impossible, for researchers to follow the relevant literature in their respective fields, even with the assistance of search tools like PubMed and Google.

The solution lies in the formalization of the interpretation presented in the scientific literature. In molecular biology, several formalisms (e.g., BEL,[87] SBML,[182] SBGN,[183] BioPAX,[184] GO-CAM)[185] have the facility to describe the interactions between biological entities that are often elucidated through laboratory or clinical experimentation. Further, there are several organizations[186–190] whose purpose is to curate and formalize the scientific literature in these formats and distribute them in one of several databases and repositories. Because curation is both difficult and time-consuming, several semi-automated NLP[191,192] curation workflows based on NLP-based relation extraction systems[193–195] and assemblers[196] have been proposed to assist.

The Biological Expression Language (BEL) captures causal, correlative, and associative relationships between biological entities along with the experimental/biological context in which they were observed as well as the provenance of the publication from which the relation was reported (https://biological-expression-languge.github.io). It uses a text-based custom domain-specific language (DSL) to enable biologists and curators alike to express the interpretations present in biomedical texts in a simple but structured form, as opposed to a complicated formalism built with low-level formats XML, JSON, and

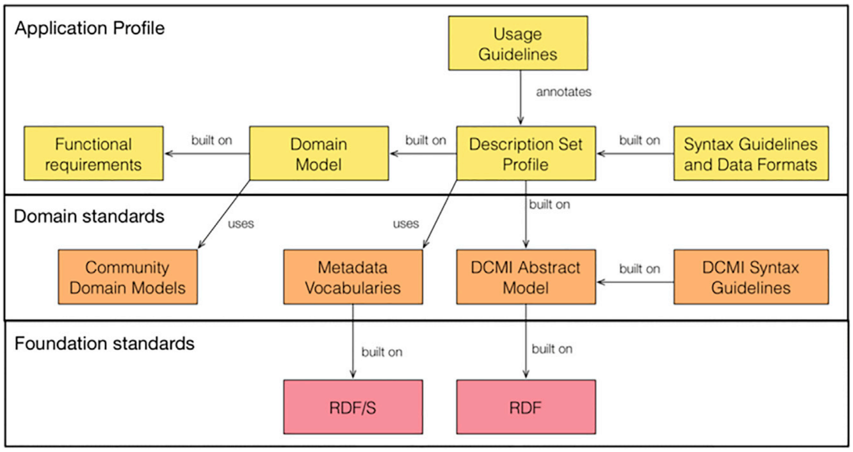


Figure 7. Singapore Framework application profile model

RDF or mid-level formats like OWL and OBO. Similarly to OWL and OBO, BEL pays deep respect to the need for the use of structured identifiers and controlled vocabularies for its statements to support the integration of multiple content sources in downstream applications. We focus on BEL because of its unique ability to represent findings across biological scales, including the genomic, transcriptomic, proteomic, pathway, phenotype, and organism levels.

Below is a representation of a portion of the MAPK signaling pathway in BEL (Box 9), which describes the process through which a series of kinases are phosphorylated, become active, and phosphorylate the next kinase in the pathway. It uses the FamPlex (fplx)[197] namespace to describe the RAF, MEK, and ERK protein families.

While the additional provenance, context, and metadata associated with each statement have not been shown, this example demonstrates that several disparate information sources can be assembled in a graph-like structure due to the triple-like nature of BEL statements.

While BEL was designed to express the interpretation presented in the literature, related formats are more focused on mechanistically describing the underlying processes on either a qualitative (e.g., BioPAX, SBGN) or quantitative (e.g., SBML) basis. Ultimately, each of these formalisms has supported a new generation of analytical techniques that have begun to replace classical pathway-analysis.

Future directions: Reproducible articles

Attempts have been made to integrate reproducible analyses into manuscripts. An article in eLife[198] was published with an inline live R Markdown Binder analysis as part of a proof-of-concept of the publisher's Executable Research Article (ERA) (Aufreiter and Penfold, 2018, IEEE eScience, abstract).[199,200] Because of the technical metadata used for rendering and display, subtle changes are required to integrate containerized analyses with JATS, and the requirements for hosting workflows outside the narrow context of Binder will require further engineering and metadata standards.

DISCUSSION

The range and diversity of metadata standards developed that aid researchers in their daily activities, also support them in sharing research outputs (data, code, publications, and other component parts of the research life cycle). If we promote metadata as the "glue" of reproducible research, what does that entail for the metadata and reproducible research communities? Clearly, no single metadata standard can support all aspects of the analytic stack. Metadata standards are driven by needs associated with function, discipline, and object format type; hence the categorization of descriptive, administrative, and structural metadata, as well as standards targeting a domain (e.g., biology) or object format (images, GIS materials).[201] The fact that metadata standards continue to undergo formal community reviews demonstrates value. Finally, as research communities and activities converge around the analytical stack and seek to automate pipelines supporting scientific services such as analytic cores, metadata not only has continuing value for reproducibility, it is shown to be critical to this endeavor.

In our review, we have attempted to describe metadata as it addresses reproducibility across the analytic stack. Two principal components: (1) embeddedness versus connectedness, and the (2) methodology weight and standardization appear to be recurring themes across all metadata facets.

Embeddedness versus connectedness

Certain efforts in the metadata context lend to the stickiness of experimental details from data collection to publications, and others are more directed to the goals of data sharing and immediate access. Data formats have an influence on the long-term reproducibility of analyses and reusability of input data, though these goals are not always aligned. Some binary data formats lend them to easily accommodate embedded metadata, i.e., metadata that is bound to its respective data by residing in the same file, not to be confused from metadata embedded in supplementary materials. In the case of the DICOM format used in medical imaging, a well-vetted set of instrumentation metadata is complemented by support for application-specific metadata. Downstream, this has enabled support for DICOM images in various repositories such as The Cancer Imaging Archive.[202] The continued increase in the use of such imaging data has led to efforts to further leverage biomedical ontologies in tags[203] and issue DOIs to individual images.[204] As discussed above, the lack of explicit support for complex metadata structures has not hindered the adoption of DICOM for a variety of uses not anticipated by its authors (DICOM introduced in 1985). This could be an argument that embeddedness is more important than complexity for long-term sustainability, or merely that early arrivals tend to stay entrenched. Software support in terms of parsing and storing embedded metadata also plays a role in the level of adoption. In the case of Binary Alignment Map[205] files used to store genomic alignments, file-level metadata resides in

Box 9. MAPK signaling pathway in Biological Expression Language

```
act(p(fplx:RAF), ma(kin)) directlyIncreases p(fplx:MEK, pmod(Ph))
  p(fplx:MEK, pmod(Ph)) directlyIncreases act(p(fplx:MEK), ma(kin))
act(p(fplx:MEK), ma(kin)) directlyIncreases  p(fplx:ERK, pmod(Ph))
  p(fplx:ERK, pmod(Ph)) directlyIncreases act(p(fplx:ERK)))
```

an optional comment section above data. Once again, these are arbitrary human-readable strings with no inherent advanced data structure capabilities. In some instances, instrumentation can aid in reproducibility by embedding crucial metadata (such as location, instrument identifiers, and various settings) in such embedded formats with no manual input, although ideally this should not simply be used at face value as a sanity check against metadata used in the analysis; for instance, to identify potential sample swaps or other integrity issues. Reliance on ad hoc formatting methods of supporting extensibility, as in through serializations using comma or semicolon delimiters, can have deleterious effects on the stability of a format. In bioinformatics, a number of genomic position-based tabular file formats have faced "last-column bloat," as new programs have piled on an increasingly diverse array of annotations.

This rigid embedded scheme employed by DICOM stands in contrast to standards such as EML, where contributors are encouraged with a flexible ontology to support supplemental metadata for the express purposes of data sharing. MIAME appears to lie somewhere in the middle, where there is a required minimal subset of tags to be supplied, much of it from the microarray instrument itself and aided by a strong open source community (Bioconductor), and paired with a data availability incentive in order to publish associated manuscripts.

In terms of reproducibility, embeddedness represents a double-edged sword. As a packaging mechanism, embedded metadata serves to preserve aspects of attribution, provenance, and semantics for the sharing of individual files, but a steadfast reliance on files can lead to siloing, which may be antithetical to discovery (the "Findable" in FAIR). Files as the sole means of research data distribution are also contrary to the recent proliferation of "microservices": Software-as-a-Service often instantiated in a serverless architecture and offering APIs. While provenance can be embedded in the headers described above, these types of files are more likely to be found at the earlier stages of an analysis, suggesting there is work to be done in developing embedded metadata solutions for notebook and report output if this is to be a viable general scheme. So much of reproducibility depends on the relay of provenance *between* layers of the analytic stack that the implementation of metadata should be optimized to encourage usage by the tools explored in this review.

Metadata is, of course, critical to the functioning of services that support the "semantic web," in which data on the world wide web is given context to enable it to be directly queried and processed, or "machine-readable." Several technologies enabling the semantic web and linked data—RDF, OWL, SKOS, SPARQL, and JSON-LD—are best recognized as metadata formats themselves or languages for metadata introspection allowing the web to behave like a database rather than a document store. Semantic web services now exist for such diverse data sources as gene-disease interactions[206] and geospatial data.[207] RDF triples are the core of knowledge graph projects such as DBpedia[208] and Bio2RDF.[209] The interest in using knowledge graphs for modeling and prediction in various domains, and the increased use of "embedding knowledge graphs," graph to vector transformations designed to augment artificial intelligence approaches,[210] has exposed the need for reproducibility and metadata standards in this area.[211]

The development of large multi-institutional data repositories that characterize "big science" and remote web services that support both remote data usage and the vision of "bringing the tools to the data" make the cloud an appealing replacement for local computing resources.[212] This dependence on data and services hosted by others, however, introduces the threat of "workflow decay"[213] that requires extensive provenance tracking or snapshotting to freeze inputs and tools in order to ensure reproducibility at a later date.

Such centralized repositories tasked with storing and distributing a quickly growing scope of metadata both in volume and complexity have also had to innovate away from rigid schemas to more flexible representations of metadata. The EMBL-EBI Biosamples portal, for example, permits user-supplied blocks of JSON to be added to samples to accommodate a wide range of metadata supporting downstream analysis.[214] These changes can, in principle, allow metadata that was previously confined to supplementary materials to be findable.

The promise of distributed annotation services, automated discovery, and the integration of disparate forms of data, using web services and thereby avoiding massive downloads, is of central import to many areas of research. However, the import of the semantic web to reproducibility is a two-sided coin. On one hand, as noted by Aranguren and Wilkinson,[215] the semantic web provides a formalized means of providing context to data, which is a crucial part of reproducibility. The semantic web is by its very nature, open, and provides a universal low barrier to data access with few dependencies other than an Internet connection. Conversely, a review of the semantic web's growing impact on cheminformatics[216] notes that issues of data integrity and provenance are of concern when steps in an analysis rely on data fetched piecemeal via a web service. The distributed nature is a challenge for reproducibility that the self-contained research compendia do not fall victim to.

Preservation

Web services of centralized repositories provide a common source reference point for several unrelated analyses, but can serve as a critical point of failure should they disappear. While archival projects such as SoftwareHeritage.org can help mitigate problems of link decay and abandonware on the software side, projects serving to provide long-term archival solutions for scientific analyses need to cache or download web service data.

Preservation refers to ensuring the creation of long-term, portable archives of analyses (sometimes referred to as "legacy workflows") that are immune to both link decay and missing dependencies. Approaches to faithfully record retrospective provenance employ either a configured or a "recorder" scheme. The configured scheme is largely characterized by such provenance modeling specifications such as CWLProv, and efforts aimed at portable analyses such as BioCompute Objects (BCO),[217] originally designed for *in silico* regulatory submissions. These generally combine Research Objects, bagit file manifests,[78] and JSON formatted specifications for capturing experimental details.

Reprozip takes a recorder approach by tracing Linux operating system calls while executing an analysis script or workflow, tracking both system dependencies and packaging them in a compressed format.[218] While this automated approach differs considerably from an entirely configured approach and doesn't negate the need for dependency management, version pinning, or containerization, it can complement these tools for reproducibility.

Methodology weight and standardization

Our review has spotlighted several metadata solutions across a spectrum of heavyweight versus lightweight solutions, bespoke versus standard solutions, and offering different levels of granularity, and adoption. Because these choices can often largely reflect those of the stakeholders involved in the design and their goals rather than immediate needs, a discussion of those groups is warranted.

Sphere of influence

Governing and standards-setting organizations (e.g., NIH, GA4GH, W3C), new applications (e.g., machine learning, translational health), new sensors (next generation sequencing, new Earth observation satellites), and trends in the greater scientific community (open science, reproducible research) are steering metadata for reproducible research in different and broader directions than traditional stakeholders, individual researchers. There are also differences in the approaches taken between different scientific fields, with the life sciences arguably more varied in both the size of projects and the level of standards than those physical sciences (e.g., LIGO). This does not discount the fact that much of the progress in metadata for reproducibility has been originally intended for other purposes, and often "bespoke," or custom-designed solutions to address the problem at hand for small labs or individual investigators. A good example is the tximeta Bioconductor package (Figure 8), which implements reference transcriptome provenance for RNA-sequencing experiments, extending a number of popular transcript quantification tools with checksum-based tracking and identification.[219] While this is an elegant solution, tximeta is focused on one analysis pattern.

In practice, concerns over reproducibility appear to be correlated with the number of stakeholders. While there are highly conscientious scientists who have built tools and standards to support the reproducibility of their own work, pressure coming from attempts to reproduce published analyses, and the heightened reuse of data among participants in multi-institution consortia, data repositories, and data commons have forced the issue.

Metadata capital and reuse

The term "metadata capital"[220] was coined to describe how an organization's efforts in producing high-quality metadata can have a positive return on investment downstream. We contend this applies to computational reproducibility. In this context, it may be useful to reposition the onus for collecting metadata along the competitiveness of smaller groups: labs, cores, and individual institutions. These smaller organizations clearly experience a reproducibility crisis in the form of impaired transfer of knowledge from outgoing to incoming trainees. However, the seminal Nature Baker survey of 1,500 scientists reported 34% of participants had not established procedures for reproducibility in their own labs.[13] Workflows like CODECHECK can ensure that enough metadata and documentation is provided for at least one person besides the author to reproduce a computational workflow once, and the experience shows that direct communication with authors is crucial to achieve that. Even the limited scope of the check, that is intentionally not formalized but relies on the codechecker's judgment, can build up metadata capital.

Metadata reuse, for replication, generalization, meta-analyses, or general research use, is enabled by explicitly designing analyses for computational reproducibility and elemental to FAIR. Reuse and extension typically demand greater metadata needs than narrow-sense reproduction, for instance, to control for batch effects or various assumptions that go into original research.[221] Often the centralized submission portals demand more expansive metadata than an individual researcher would anticipate being necessary, belying their importance in the reproducibility and reuse process. In a similar vein, smaller domain data repositories (e.g., PANGAEA, https://www.pangaea.de) have both much higher requirements for metadata and the experienced staff to curate this metadata than general purpose repositories (e.g., Zenodo). Essential for reproducibility, surveys suggest provenance information is an important criteria for reuse in the physical sciences.[222] Metadata for data reuse has relevance for data harmonization for biomedical applications, such as toward highly granular phenotype ontologies, genotype-phenotype meta-analyses,[223] generating synthetic controls for clinical trials, and consent metadata, such as the Data Use Ontology,[224] to describe allowed downstream usage of patient data. Designing metadata for the needs of general reuse, especially outside narrow scientific domains, may require greater foresight than that needed for reproducibility but authors can follow similar templates.

Recommendations and future work

Widespread adoption of reproducible computational research is highly dependent on a cultural shift within the scientific community[225,226] promoted by both journals and funding agencies. The allegorical "stick" of higher reproducibility standards should be accompanied by carrots in the form of publication incentives. For example, the *Journal of Water Resources Planning and Management* waives or reduces fees of reproducible papers,[227] and several journals promote reproducible works, e.g., through badges.[228] Other carrots could involve a support mechanism by which pre- and post-publication peer review can properly evaluate and test statistical methods cited in papers. Such a collaborative computational peer review could involve parameter exploration, swapping out individual statistical tests or tool components for similar substitutes, and using new datasets. We envision this "reproducibility-enabled advocated software peer review" could be conducted by reviewers taking a hands-on approach to strengthening analyses using collaborative interactive notebooks or other tools.[229] The recognition of

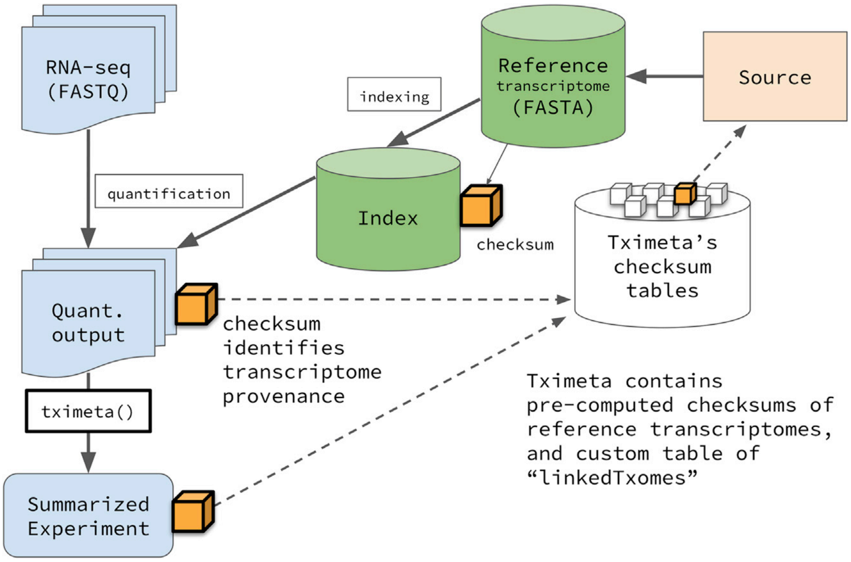

RNA-seq (FASTQ)
indexing
Reference transcriptome (FASTA)
Source
quantification
Index
checksum
Tximeta's checksum tables
Quant. output
checksum identifies transcriptome provenance
tximeta()
Tximeta contains pre-computed checksums of reference transcriptomes, and custom table of "linkedTxomes"
Summarized Experiment

Figure 8. tximeta
The high-level schematic of tximeta.[219]

sharing reproducible works, the reproductions, and more extensive reviewing efforts as important scientific contributions are paramount for their adoption and reproducibility practices will not be established without tackling the shortcomings of researcher evaluation culture.[230] Software citation[231] and software publications[232] are concrete mechanisms to value tools for reproducible computational research, and these mechanisms need higher uptake beyond disciplines that are naturally close to software. Only then can related metadata reach a broad adoption via generally used research software.

One interesting development in the area of incentives is the growing interest in developing FAIR metrics and reproducibility "badges" to denote compliance. The FAIRshake toolkit implements rubrics to evaluate the digital resources such as datasets, tools, and workflows.[233] These rubrics include criteria such as data and code availability but also metadata such as contact information, description, and licensing embedded using Schema.org tags.

Another important trend is the emergence of reporting guidelines, essentially checklists, many of which are found in the EQUATOR network,[234] perhaps the most prominent being CONSORT (Consolidated Standards of Reporting Trials) originally from 1996 but updated in 2010.[235] Newer examples include STORMS (Strengthening The Organization and Reporting of Microbiome Studies) and STROBE (Strengthening the Reporting of Observational Studies in Epidemiology). Such guidelines, while useful for authors, are rarely paired with metadata schema to allow them to be machine-readable. We envision that some of these templates may eventually be auto-generated from computational workflows.

In terms of the analytic stack, there are several areas that offer low-hanging fruit for innovation. One is developing inline semantic metadata for publications and notebooks. While Schema.org tags have been used for indexing data, to our knowledge there is no journal that supports, much less encourages, semantic markup of specific terms within a manuscript. There has been tacit support for such inline markup in newer manuscript composition tools such as Manubot[236] or the ERC's bindings,[170] but generally such terms could disambiguate concepts, point to the provenance of findings within a result section or from a figure, accelerate linked data and discovery, and improve understandability. Despite their significance, funding for the sustainable development and maintenance of core tools for research is scarce.[237] Resources for the continued improvement of software, such as the Chan Zuckerberg Initiative's "Essential Open Source Software for Science" program (https://chanzuckerberg.com/rfa/essential-open-source-software-for-science/) or NumFOCUS (https://numfocus.org) may be poised to improve the reproducibility infrastructure more than proof-of-concept tools.

The lack of integration between notebooks and pipeline frameworks can create friction during the analysis process, which can discourage users from using them jointly. Efforts such as NoWorkflow/YesWorkflow and internal frameworks such as Targets[238] are helping to bridge these distinctions, but few solutions have sought to aid notebook-pipeline integration in general.

Furthermore, there is a clear need for greater annotation within statistical reports and notebooks for semantic markup to categorize and disambiguate machine learning and deep learning workflows. Because of the explosion in advances from this area, researchers outside the machine learning core community have found it difficult to keep up with the litany of terminology, techniques, and metrics being developed. Clearly metadata can play a role in augmenting understanding of, for instance, how an existing technique relates most closely with a new one. In addition, the data management tools frequently have built-in metadata templates guiding researchers, and even applications for automatically generating the metadata. And services, such as Dryad, have front-line curators offering quality control. This will facilitate the broader goals of reproducibility.

Statistical metadata are vital for users to discover, and reviewers to evaluate complex statistical analyses,[239] but metadata that describes statistical methods is largely nonexistent. The increasing diversity and application of machine learning approaches makes it increasingly difficult to discern the intent and provenance of statistical methods.

This confusion has serious consequences for the peer review system, as it provides more opportunities for submitters to engage in "p-hacking," cherry-picking algorithms and parameters that return a desired level of significance. Another, perhaps less common, tactic is "steamrolling" reviewers by submitting a novel, opaque algorithm to support a scientific hypothesis. Without reproducible code, evaluating such submissions becomes impossible. Both of these strategies are arrested by reproducible research standards at the publication level.

To test the robustness of a set of results, reviewers should be able to swap in similar methods, but identifying and actually applying an equivalent statistical method is not for the weak

Box 10. Glossary

benchmark: a comparison of existing tools and models using a gold standard and a limited number of metrics
big data: data that are too large, too complex, too dynamic, too varying, or too unstructured to be analyzed with regular statistical methods
computing environment: the totality of hard- and software components involved in a particular scientific workflow, e.g., versions of used software, make and model of used processor; can be documented for both machines and humans
computational workflow: using algorithms and statistical methods to load, transform, analyze, and visualize digital data
container: a lightweight machine computing environment that serves to isolate processes and provide a compatible virtual operating system interface for individual software applications
dependency manager: tools that rely structured files with dependencies, including versions or installation source, to automatically provision all dependencies of a specific software for use or development
Docker: the most popular containerization scheme (see ‘‘container’’)
Domain-specific language: a collection of specialized operators that resembles a general programming language but is designed to handle a very specific task
Generalizable: analysis that can accommodate changes to both code and data while maintaining core findings
Javascript object notation (JSON): simple file format with support for a data types such as arrays and dictionaries
literate programming: computer code and documentation are interspersed in a single source file, which can be compiled or ‘‘woven’’ by tools into other formats for reading, exposing, or hiding parts of the code
Web Ontology Language (OWL): provides syntax for describing allowed entities and their allowed relations
pipeline framework: software and associated languages that provide a means of abstracting and executing reentrant and distributed computational operations
PROV: a W3C ontology providing the basic concepts of agents, entities, and activities used by a number of reproducibility standards
questionable research practices (QRP): practices that compromise scientific integrity such as hypothesizing after the results are known, selective reporting, and inflation bias
Resource description framework (RDF): data model composed of entity-attribute-value triplets
registered report: a study in which a data collection and analysis plan have been pre-registered and reviewed prior to commencement
replicable: returning core findings using a different dataset
reproducible computational research (RCR): area of reproducible research concerned mainly with computational, or *in silico* analyses, as opposed to wet-lab, field studies, or other scientific methods in the physical world
research compendium: collection of files that accompanies, enhances a scientific publication by providing data, code, and documentation for reproducing a computational workflow
retrospective provenance: a record of an executed task or workflow, i.e., what was actually run
prospective provenance: a plan of execution described by a workflow, i.e., what will be run
robustness (tool): the ability of software application or model to accommodate different datasets and experimental designs and return tenable results
robustness (reproducibility): the ability of *in silico* study or analysis to accommodate roughly equivalent tool substitutions and return similar results
runnability: ability for computational analysis to be packaged in a way that end-users can quickly achieve arrive at results from raw data
software dependency: one software relies on features from another software
truth challenge: a competitive benchmark with a training set and sequestered test sets

of heart. As an example, consider gradient boosted trees, a method of building and improving predictive models that involves weak learners (classifiers only slightly better than random guess) using decision trees. Random forests is a popular machine learning algorithm for classification, also decision tree–based. The choice between these two methods is so subtle that even experienced data scientists may have to evaluate them empirically but may substantially change model predictions given limited data. As such, the test of robustness is the flip side of the coin from benchmarking exercise.[240] In a test of robustness, the analysis itself, borne from the scientific hypothesis, is under scrutiny, while in a benchmark the tools or models are under scrutiny.

Metadata standards that can support lightweight and heavyweight solutions are well positioned for sustainability and adoption, as are those that provide connections between layers of the analytic stack without a steep learning curve or are fully integrated into tools and thereby invisible to the user. One example of this, which to our knowledge has yet not been implemented, is file format and content sanity checks defined by input metadata but implemented at the pipeline level. The capturing and documentation of the computing environment, hardware and software, also requires conscious steps by authors and could be largely automated if a common metadata standard existed. For practical reproducible computational research, these metadata on tools are just as important as the data and code itself.

Finally there needs to be greater emphasis on translation between embedded and distributed metadata solutions and the automation of meaningful metadata creation. As discussed, files that support embedded metadata excel as data currency, but may not be ideal for warehousing, querying, or remote access. Conversely, solutions that rely on databases for metadata storage to offer advanced features, whether they be for input metadata, provenance tracking, or workflow execution, usually do so at the expense of portability. Systems and standards that provide conduits between these realities are more likely to succeed.

While metadata will always serve as the "who, what, where, why, and how" of data, it is also increasingly the mechanism by which scientific output is made reusable and useful. In our review, we have attempted to highlight reproducibility as a vital formal area of metadata research and underscore metadata as an indispensable facet of reproducible computational research.

Software and resource availability

A repository containing metadata and examples for the standards discussed in this paper, as well as code for reproducing the figures, is available at https://github.com/leipzig/metadata-in-rcr. A glossary containing important terms can be found below (Box 10).

ACKNOWLEDGMENTS

We thank David Clunie, Jian Qin, and Farah Zaib Khan, and the five anonymous reviewers for their helpful comments and suggestions. Stian Soiland-Reyes contributed to an earlier version of this article.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Margolis, R., Derr, L., Dunn, M., Huerta, M., Larkin, J., Sheehan, J., Guyer, M., and Green, E.D. (2014). The National Institutes of Health's big data to knowledge (BD2K) initiative: capitalizing on biomedical big data. J. Am. Med. Inform. Assoc. *21*, 957–958.

2. Brito, J.J., Li, J., Moore, J.H., Greene, C.S., Nogoy, N.A., Garmire, L.X., and Mangul, S. (2020). Recommendations to enhance rigor and reproducibility in biomedical research. Gigascience *9*. https://doi.org/10.1093/gigascience/giaa056.

3. National Academies of Sciences, Engineering, and Medicine; Policy and Global Affairs; Committee on Science, Engineering, Medicine, and Public Policy; Board on Research Data and Information; Division on Engineering and Physical Sciences; Committee on Applied and Theoretical Statistics; Board on Mathematical Sciences and Analytics; Division on Earth and Life Studies; Nuclear and Radiation Studies Board; Division of Behavioral and Social Sciences and Education; Committee on National Statistics; Board on Behavioral, Cognitive, and Sensory Sciences (2019). Committee on Reproducibility and Replicability in Science. Reproducibility and Replicability in Science (National Academies Press).

4. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L.B., Bourne, P.E., et al. (2016). The FAIR guiding principles for scientific data management and stewardship. Sci. Data *3*, 160018.

5. Leipzig, J. (2019). Awesome Reproducible Research.

6. Donoho, D.L. (2010). An invitation to reproducible computational research. Biostatistics *11*, 385–388.

7. Li, K., Greenberg, J., and Dunic, J. (2020). Data objects and documenting scientific processes: an analysis of data events in biodiversity data papers. J. Assoc. Inf. Sci. Technol. *71*, 172–182.

8. Stodden, V., Borwein, J., and Bailey, D.H. (2013). Setting the Default to Reproducible. Computat. Sci. Res. *46*, 4–6.

9. Whitaker, K. (2016). Showing your working: a guide to reproducible neuroimaging analyses. Figshare. https://doi.org/10.6084/m9.figshare.4244996.v1.

10. The Alan Turing Institute (2019). The Turing Way: A Handbook for Reproducible Data Science. https://www.turing.ac.uk/research/research-projects/turing-way-handbook-reproducible-data-science.

11. Barba, L.A. (2018). Terminologies for reproducible research. arXiv.

12. Association for Computing Machinery (2020). New Changes to Badging Terminology. https://www.acm.org/publications/badging-terms.

13. Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. Nat. News *533*, 452.

14. Baggerly, K. (2010). Disclose all data in publications. Nature *467*, 401.

15. Begley, C.G., and Ellis, L.M. (2012). Drug development: raise standards for preclinical cancer research. Nature *483*, 531–533.

16. Ioannidis, J.P.A., Allison, D.B., Ball, C.A., Coulibaly, I., Cui, X., Culhane, A.C., Falchi, M., Furlanello, C., Game, L., Jurman, G., et al. (2009). Repeatability of published microarray gene expression analyses. Nat. Genet. *41*, 149–155.

17. Motulsky, H.J. (2014). Common misconceptions about data analysis and statistics. J. Pharmacol. Exp. Ther. *351*, 200–205.

18. Ioannidis, J.P.A. (2005). Why most published research findings are false. PLoS Med. *2*, e124.

19. Fanelli, D. (2018). Opinion: Is science really facing a reproducibility crisis, and do we need it to? Proc. Natl. Acad. Sci. U S A *115*, 2628–2631.

20. Obels, P., Lakens, D., Coles, N.A., Gottfried, J., and Green, S.A. (2020). Analysis of open data and computational reproducibility in registered reports in psychology. Advances in Methods and Practices in Psychological Science *3*, 229–237.

21. Lehrer, J. (2010). The truth wears off. New Yorker *13*, 229.

22. Greenberg, J., Swauger, S., and Feinstein, E. (2013). Metadata capital in a data repository. In International Conference on Dublin Core and Metadata Applications, pp. 140–150. https://dcpapers.dublincore.org/pubs/article/view/3678.

23. Rousidis, D., Garoufallou, E., Balatsoukas, P., and Sicilia, M.-A. (2014). Metadata for big data: a preliminary investigation of metadata quality issues in research data repositories. Inf. Serv. Use *34*, 279–286.

24. Ekbia, H., Mattioli, M., Kouper, I., Arave, G., Ghazinejad, A., Bowman, T., Suri, V.R., Tsou, A., Weingart, S., and Sugimoto, C.R. (2015). Big data, bigger dilemmas: a critical review. J. Assn Inf. Sci. Tec *66*, 1523–1545.

25. Warden, P. (2018). The machine learning reproducibility crisis. https://petewarden.com/2018/03/19/the-machine-learning-reproducibility-crisis/.

26. Bouthillier, X., Laurent, C., and Vincent, P. (2019). Unreproducible research is reproducible. In Proceedings of the 36th International Conference on Machine Learning, *97*, K. Chaudhuri and R. Salakhutdinov, eds., pp. 725–734.

27. Schelter, S., Boese, J.-H., Kirschnick, J., Klein, T., and Seufert, S. (2017). Automatically tracking metadata and provenance of machine learning experiments. In Machine Learning Systems Workshop at NIPS http://learningsys.org/nips17/assets/papers/paper_13.pdf.

28. Rauh, S., Torgerson, T., Johnson, A.L., Pollard, J., Tritz, D., and Vassar, M. (2020). Reproducible and transparent research practices in published neurology research. Res. Integr. Peer Rev. *5*, 5.

29. Stodden, V., Krafczyk, M.S., and Bhaskar, A. (2018). Enabling the verification of computational results: an empirical evaluation of computational reproducibility. In Proceedings of the First International Workshop on Practical Reproducible Evaluation of Computer Systems; P-RECS'18, I. Jimenez, C. Maltzahn, and J. Lofstead, eds. (Association for Computing Machinery), pp. 1–5.

30. Stagge, J.H., Rosenberg, D.E., Abdallah, A.M., Akbar, H., Attallah, N.A., and James, R. (2019). Assessing data availability and research reproducibility in hydrology and water resources. Sci. Data *6*, 190030.

31. Nüst, D., Granell, C., Hofer, B., Konkol, M., Ostermann, F.O., Sileryte, R., and Cerutti, V. (2018). Reproducible research and GIScience: an evaluation using AGILE conference papers. PeerJ *6*, e5072.

32. Sandve, G.K., Nekrutenko, A., Taylor, J., and Hovig, E. (2013). Ten simple rules for reproducible computational research. PLoS Comput. Biol. *9*, e1003285.

33. Collberg, C., Proebsting, T., Moraila, G., Shankaran, A., Shi, Z., and Warren, A.M. (2014). Measuring Reproducibility in Computer Systems Research (Department of Computer Science, University of Arizona).

34. Piccolo, S.R., and Frampton, M.B. (2016). Tools and techniques for computational reproducibility. Gigascience *5*, 30.

35. FitzJohn, R., Pennell, M., Zanne, A., and Cornwell, W. (2014). Reproducible research is still a challenge. rOpenSci https://ropensci.org/blog/2014/06/09/reproducibility/#:~:text=Science%20is%20reportedly%20in%20the%20middle%20of%20a%20reproducibility%20crisis.&text=In%20general%20the%20argument%20is,that%20cannot%20be%20independently%20reproduced.

36. Ball, A. (2009). Scientific Data Application Profile Scoping Study Report. http://www.ukoln.ac.uk/projects/sdapss/.

37. Ball, A., Greenberg, J., Jeffery, K., and Koskela, R. (2016). RDA Metadata Standards Directory Working Group: Final Report (Research Data Alliance).

38. Riley, J. (2017). Understanding Metadata: What is Metadata, and What is it For?: A Primer (National Information Standards Organization). http://www.niso.org/publications/understanding-metadata-2017.

39. Qin, J., and Zeng, M. (2016). Metadata (ALA-Neal Schuman).

40. Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C.A., Causton, H.C., et al. (2001). Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. Nat. Genet. *29*, 365–371.

41. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Michael Cherry, J., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al. (2000). Gene ontology: tool for the unification of biology. Nat. Genet. *25*, 25–29.

42. Pearson, W.R. (1990). Rapid and sensitive sequence comparison with FASTP and FASTA. Methods Enzymol. *183*, 63–98.

43. Wilkinson, M.D., Sansone, S.-A., Schultes, E., Doorn, P., Bonino da Silva Santos, L.O., and Dumontier, M. (2018). A design framework and exemplar metrics for FAIRness. Sci. Data *5*, 180118.

44. Paskin, N. (2010). Digital object identifier (DOI®) system. Encyclopedia Libr. Inf. Sci. *3*, 1586–1592.

45. Sansone, S.-A., McQuilton, P., Rocca-Serra, P., Gonzalez-Beltran, A., Izzo, M., Lister, A.L., and Thurston, M.; FAIRsharing Community (2019). FAIRsharing as a community approach to standards, repositories and policies. Nat. Biotechnol. *37*, 358–367.

46. Qin, J., Dobreski, B., and Brown, D. (2016). Metadata and reproducibility: a case study of gravitational wave research data management. Int. J. Digital Curation *11*, 218–231.

47. Page, K., Palma, R., Holubowicz, P., Klyne, G., Soiland-Reyes, S., Cruickshank, D., Cabero, R.G., Cuesta, E.G., De Roure, D., Zhao, J., et al. (2012). From workflows to research objects: an architecture for preserving the semantics of science. In Proceedings of the 2nd International Workshop on Linked Science https://www.semanticscholar.org/paper/From-Workflows-to-Research-Objects%3A-An-Architecture-Page-Palma/a1c9e4c4a7c6a552075c08fa6efd26d33c664096.

48. Wirth, R., and Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. . Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining (Springer-Verlag London), pp. 29–39.

49. Lenhardt, W., Ahalt, S., Blanton, B., Christopherson, L., and Idaszak, R. (2014). Data management lifecycle and software lifecycle management in the context of conducting science. J. Open Res. Softw. *2*, e15.

50. Michener, W.K. (2006). Meta-information concepts for ecological data management. Ecol. Inform. *1*, 3–7.

51. Bidgood, W.D., Jr., and Horii, S.C. (1992). Introduction to the ACR-NEMA DICOM standard. Radiographics *12*, 345–355.

52. Robertson, T., Döring, M., Guralnick, R., Bloom, D., Wieczorek, J., Braak, K., Otegui, J., Russell, L., and Desmet, P. (2014). The GBIF integrated publishing toolkit: facilitating the efficient publishing of biodiversity data on the internet. PLoS One *9*, e102623.

53. Bernstein, H.J., Bollinger, J.C., Brown, I.D., Gražulis, S., Hester, J.R., McMahon, B., Spadaccini, N., Westbrook, J.D., and Westrip, S.P. (2016). Specification of the crystallographic information file format, version 2.0. J. Appl. Crystallogr. *49*, 277–284.

54. Chirico, R.D., Frenkel, M., Diky, V.V., Marsh, K.N., and Wilhout, R.C. (2003). ThermoML an XML-based approach for storage and exchange of experimental and critically evaluated thermophysical and thermochemical property data. 2. Uncertainties. J. Chem. Eng. Data *48*, 1344–1359.

55. Cuellar, A.A., Lloyd, C.M., Nielsen, P.F., Bullivant, D.P., Nickerson, D.P., and Hunter, P.J. (2003). An Overview of CellML 1.1, a biological model description language. Simulation *79*, 740–747.

56. Alter, G., Gonzalez-Beltran, A., Ohno-Machado, L., and Rocca-Serra, P. (2020). The data tags suite (DATS) model for discovering data access and use requirements. Gigascience *9*, giz165. https://doi.org/10.1093/gigascience/giz165.

57. Andersson, L., Archibald, A.L., Bottema, C.D., Brauning, R., Burgess, S.C., Burt, D.W., Casas, E., Cheng, H.H., Clarke, L., Couldrey, C., et al. (2015). Coordinated International action to accelerate genome-to-phenome with FAANG, the functional annotation of animal genomes project. Genome Biol. *16*, 57.

58. International Organization for Standardization. ISO/TC 276 – Biotechnology. https://www.iso.org/committee/4514241.html.

59. Ison, J., Rapacki, K., Ménager, H., Kalaš, M., Rydza, E., Chmura, P., Anthon, C., Beard, N., Berka, K., Bolser, D., et al. (2016). And data services registry: a community effort to document bioinformatics resources. Nucleic Acids Res. *44*, D38–D47.

60. Anaconda. Defining Metadata (meta.Yaml) — Conda-Build 3.19.3+29.gba6cf7ab.Dirty Documentation. https://docs.conda.io/projects/conda-build/en/latest/resources/define-metadata.html.

61. Dumbill, E. (2010). DOAP: description of a project. https://github.com/ewilderj/doap/wiki.

62. Gil, Y., Ratnakar, V., and Garijo, D. (2015). OntoSoft: capturing scientific software metadata. In Proceedings of the 8th International Conference on Knowledge Capture, K. Barker and J.M. Gómez-Pérez, eds. (Association for Computing Machinery), p. 32.

63. Ison, J., Kalas, M., Jonassen, I., Bolser, D., Uludag, M., McWilliam, H., Malone, J., Lopez, R., Pettifer, S., and Rice, P. (2013). EDAM: An ontology of bioinformatics operations, types of data and identifiers, topics and formats. Bioinformatics *29*, 1325–1332.

64. Malone, J., Brown, A., Lister, A.L., Ison, J., Hull, D., Parkinson, H., and Stevens, R. (2014). The software ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation. J. Biomed. Semantics *5*, 25.

65. Zheng, J., Harris, M.R., Masci, A.M., Lin, Y., Hero, A., Smith, B., and He, Y. (2016). The ontology of biological and clinical statistics (OBCS) for standardized and reproducible statistical analysis. J. Biomed. Semantics *7*, 53.

66. (2016). STATO: An Ontology of Statistical Methods. http://stato-ontology.org/.

67. Capadisli, S., Auer, S., and Ngonga Ngomo, A.-C. (2015). Linked SDMX data. Semantic Web *6*, 105–112.

68. Hoyle, L., and Wackerow, J. (2016). DDI as a Common Format for Export and Import for Statistical Packages. IASSIST Quarterly *39*.

69. Esteves, D., Moussallem, D., Neto, C.B., Soru, T., Usbeck, R., Ackermann, M., and Lehmann, J. (2015). MEX Vocabulary: A lightweight Interchange format for machine learning experiments. In Proceedings of the 11th International Conference on Semantic Systems; SEMANTICS '15, S. Hellmann, J.X. Parreira, and A. Polleres, eds. (Association for Computing Machinery), pp. 169–176.

70. Publio, G.C., Esteves, D., Ławrynowicz, A., Panov, P., Soldatova, L., Soru, T., Vanschoren, J., and Zafar, H. (2018). ML-schema: exposing the semantics of machine learning with schemas and ontologies. arXiv.

71. Peter, A., Michael, R.,C., Nebojša, T., Brad, C., John, C., Michael, H., Andrey, K., Dan, L., Hervé, M., Maya, N., et al. (2016). Common Workflow Language, v1.0 (Figshare). https://doi.org/10.6084/m9.figshare.3115156.v2.

72. Santana-Perez, I., Ferreira da Silva, R., Rynge, M., Deelman, E., Pérez-Hernández, M.S., and Corcho, O. (2017). Reproducibility of execution environments in computational science using semantics and clouds. Future Gener. Comput. Syst. *67*, 354–367.

73. Ding, L., Futrelle, J., Garijo, D., Groth, P., Jewell, M., Miles, S., Missier, P., Pan, J., and Zhao, J. (2010). Open Provenance Model (OPM) OWL Specification, L. Moreau, ed..

74. Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., et al. (2013). PROV-O: The PROV Ontology (W3C Recommendation, World Wide Web Consortium). http://www.w3.org/TR/prov-o/.

75. Khan, F.Z., Soiland-Reyes, S., Sinnott, R.O., Lonie, A., Goble, C., and Crusoe, M.R. (2019). Sharing interoperable workflow provenance: a review of best practices and their practical application in CWLProv. Gigascience *8*. https://doi.org/10.1093/gigascience/giz095.

76. Cao, Y., Jones, C., Cuevas-Vicenttín, V., Jones, M.B., Ludäscher, B., McPhillips, T., Missier, P., Schwalm, C., Slaughter, P., Vieglais, D., et al. (2016). ProvONE: extending PROV to support the DataONE scientific community.

77. Ciccarese, P., Soiland-Reyes, S., Belhajjame, K., Gray, A.J., Goble, C., and Clark, T. (2013). PAV ontology: provenance, authoring and versioning. J. Biomed. Semantics *4*, 37.

78. Kunze, J., Littman, J., Madden, E., Scancella, J., and Adams, C. (2018). The BagIt File Packaging Format (V1.0). RFC Editor. https://doi.org/10.17487/RFC8493.

79. Alterovitz, G., Dean, D., Goble, C., Crusoe, M.R., Soiland-Reyes, S., Bell, A., Hayes, A., Suresh, A., Purkayastha, A., King, C.H., et al. (2018). Enabling precision medicine via standard communication of HTS provenance, analysis, and results. PLoS Biol. *16*, e3000099.

80. Weibel, S., Kunze, J., Lagoze, C., and Wolf, M. (1998). Dublin core metadata for resource discovery. Internet Eng. Task Force RFC *2413*, 132.

81. Huh, S. (2014). Journal article tag suite 1.0: National information standards organization standard of journal extensible markup language. Sci. Ed. *1*, 99–104. https://doi.org/10.6087/kcse.2014.1.99.

82. Needleman, M.H. (2001). ONIX (online information exchange). Serials Rev. *27*, 102–104.

83. Lipscomb, C.E. (2000). Medical subject headings (MeSH). Bull. Med. Libr. Assoc. *88*, 265–266.

84. Chan, L.M. (1995). Library of Congress Subject Headings: Principles and Application, Third Edition (Libraries Unlimited).

85. Clark, T., Ciccarese, P.N., and Goble, C.A. (2014). Micropublications: a semantic model for claims, evidence, arguments and annotations in biomedical communications. J. Biomed. Semantics *5*, 28.

86. Williams, A.J., Harland, L., Groth, P., Pettifer, S., Chichester, C., Willighagen, E.L., Evelo, C.T., Blomberg, N., Ecker, G., Goble, C., and Mons, B. (2012). Open PHACTS: Semantic interoperability for drug discovery. Drug Discov. Today *17*, 1188–1198.

87. Slater, T. (2014). Recent advances in modeling languages for pathway maps and computable biological networks. Drug Discov. Today *19*, 193–198.

88. Ciccarese, P., Wu, E., Wong, G., Ocana, M., Kinoshita, J., Ruttenberg, A., and Clark, T. (2008). The SWAN biomedical discourse ontology. J. Biomed. Inform. *41*, 739–751.

89. Peroni, S. (2014). The semantic publishing and referencing ontologies. In Semantic Web Technologies and Legal Scholarly Publishing, S. Peroni, ed. (Springer International Publishing), pp. 121–193.

90. Gangemi, A., Peroni, S., Shotton, D., and Vitali, F. (2017). Semantic Web*0* (The Publishing Workflow Ontology (PWO)), pp. 1–12.

91. Peng, K., Safonova, Y., Shugay, M., Popejoy, A.B., Rodriguez, O.L., Breden, F., Brodin, P., Burkhardt, A.M., Bustamante, C., Cao-Lormeau, V.-M., et al. (2021). Diversity in Immunogenomics: the value and the challenge. Nat. Methods *18*, 588–591.

92. Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L.J., Eilbeck, K., Ireland, A., Mungall, C.J., et al. (2007). The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nat. Biotechnol. *25*, 1251–1255.

93. Graham, R.N.J., Perriss, R.W., and Scarsbrook, A.F. (2005). DICOM demystified: a review of digital file formats and their use in radiological practice. Clin. Radiol. *60*, 1133–1140.

94. Whitcher, B., Schmid, V.J., and Thornton, A. (2011). Working with the DICOM and NIfTI data standards in R. J. Stat. Softw. *44*. https://doi.org/10.18637/jss.v044.i06.

95. Gueld, M.O., Kohnen, M., Keysers, D., Schubert, H., Wein, B.B., Bredno, J., and Lehmann, T.M. (2002). Quality of DICOM header information for image categorization. In Medical Imaging 2002: PACS and Integrated Medical Information Systems: Design and Evaluation, *4685*, E.L. Siegel and H.K. Huang, eds. (International Society for Optics and Photonics), pp. 280–287.

96. Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.-C., Pujol, S., Bauer, C., Jennings, D., Fennessy, F., Sonka, M., et al. (2012). 3D slicer as an image computing platform for the quantitative imaging network. Magn. Reson. Imaging *30*, 1323–1341.

97. Herz, C., Fillion-Robin, J.-C., Onken, M., Riesmeier, J., Lasso, A., Pinter, C., Fichtinger, G., Pieper, S., Clunie, D., Kikinis, R., et al. (2017). Dcmqi: An open source library for standardized communication of quantitative image analysis results using DICOM. Cancer Res. *77*, e87–e90.

98. Halpern, B., Frazier, M., Potapenko, J., Casey, K., Koenig, K., Longo, C., Lowndes, J.S., Rockwood, C.R., Setig, E., Selkoe, K., et al. (2015). Cumulative Human Impacts: raw stressor data (2008 and 2013). KNB, 10.5063/F1S180FS.

99. Faith, J.J., Driscoll, M.E., Fusaro, V.A., Cosgrove, E.J., Hayete, B., Juhn, F.S., Schneider, S.J., and Gardner, T.S. (2008). Many microbe microarrays database: uniformly normalized affymetrix compendia with structured experimental metadata. Nucleic Acids Res. *36*, D866–D870.

100. Ramasamy, A., Mondry, A., Holmes, C.C., and Altman, D.G. (2008). Key issues in conducting a meta-analysis of gene expression microarray datasets. Plos Med. *5*, e184.

101. Rocca-Serra, P., Brandizi, M., Maguire, E., Sklyar, N., Taylor, C., Begley, K., Field, D., Harris, S., Hide, W., Hofmann, O., et al. (2010). ISA software suite: supporting standards-compliant experimental annotation and enabling curation at the community level. Bioinformatics *26*, 2354–2356.

102. Pezoa, F., Reutter, J.L., Suarez, F., Ugarte, M., and Vrgoč, D. (2016). Foundations of JSON schema. In Proceedings of the 25th International Conference on World Wide Web (International World Wide Web Conferences Steering Committee), pp. 263–273.

103. Janowicz, K., Hitzler, P., Adams, B., Kolas, D., and Vardeman, C. (2014). Five stars of linked data vocabulary use. Semantic Web *5*, 173–176.

104. Brickley, D., Burgess, M., and Noy, N. (2019). Google dataset search: building a search engine for datasets in an open web ecosystem. In The World Wide Web Conference (Association for Computing Machinery), pp. 1365–1375.

105. Pérignon, C., Gadouche, K., Hurlin, C., Silberman, R., and Debonnel, E. (2019). Certify reproducibility with confidential data. Science *365*, 127–128.

106. Foster, I. (2018). Research infrastructure for the safe analysis of sensitive data. Ann. Am. Acad. Pol. Soc. Sci. *675*, 102–120.

107. Jaradeh, M.Y., Oelen, A., Farfar, K.E., Prinz, M., D'Souza, J., Kismihók, G., Stocker, M., and Auer, S. (2019). Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. https://arxiv.org/abs/1901.10816.

108. Rajesh, A., Chang, Y., Abedalthagafi, M.S., Wong-Beringer, A., Love, M.I., and Mangul, S. (2021). Improving the completeness of public metadata accompanying omics studies. Genome Biol. *22*, 106.

109. Papoutsoglou, E.A., Faria, D., Arend, D., Arnaud, E., Athanasiadis, I.N., Chaves, I., Coppens, F., Cornut, G., Costa, B.V., Ćwiek-Kupczyńska, H., et al. (2020). Enabling reusability of plant phenomic datasets with MIAPPE 1.1. New Phytol. *227*, 260–273.

110. Oberkampf, H., Krieg, H., Senger, C., Weber, T., and Colsman, W. (2018). 20 Allotrope Data Format – Semantic Data Management in Life Sciences. https://doi.org/10.6084/m9.figshare.7346489.v1.

111. Stathias, V., Koleti, A., Vidović, D., Cooper, D.J., Jagodnik, K.M., Terryn, R., Forlin, M., Chung, C., Torre, D., Ayad, N., et al. (2018). Sustainable data and metadata management at the BD2K-LINCS data coordination and integration center. Sci. Data *5*, 180117.

112. Love, M.I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol. *15*, 550.

113. Beaulieu-Jones, B.K., and Greene, C.S. (2017). Reproducibility of computational workflows is automated using continuous analysis. Nat. Biotechnol. *35*, 342–346.

114. Palmblad, M., Lamprecht, A.-L., Ison, J., and Schwämmle, V. (2019). Automated workflow composition in mass spectrometry-based proteomics. Bioinformatics *35*, 656–664.

115. Hillion, K.-H., Kuzmin, I., Khodak, A., Rasche, E., Crusoe, M., Peterson, H., Ison, J., and Ménager, H. (2017). Using Bio.tools to generate and annotate workbench tool descriptions. F1000Res. *6*. https://doi.org/10.12688/f1000research.12974.1.

116. Bedő, J. (2019). BioShake: A haskell EDSL for bioinformatics workflows. PeerJ *7*, e7223.

117. Amstutz, P., Tijanić, N., Soiland-Reyes, S., Kern, J., Stojanovic, L., Pierce, T., et al. (2015). Portable workflow and tool descriptions with the CWL. In Bioinformatics Open Source Conference, N. Harris and P. Cock, eds. (Open Bioinformatics Foundation).

118. Kumar, A., Rasche, H., Grüning, B., and Backofen, R. (2021). Tool recommender system in Galaxy using deep learning. Gigascience *10*. https://doi.org/10.1093/gigascience/giaa152.

119. Jones, M.B., Boettiger, C., Mayes, A.C., Smith, A., Slaughter, P., Niemeyer, K., Gil, Y., Fenner, M., Nowak, K., Hahnel, M., et al. (2016). CodeMeta: an exchange schema for software metadata. KNB Data Repository. https://doi.org/10.5063/schema/codemeta-1.0.

120. Price, M.H. (2020). Baydem (Github).

121. Smith, A.M., Katz, D.S., and Niemeyer, K.E. (2016). Software citation principles. Peerj Comput. Sci. *2*, e86.

122. Wattanakriengkrai, S., Chinthanet, B., Hata, H., Kula, R.G., Treude, C., Guo, J., and Matsumoto, K. (2020). GitHub repositories with links to academic papers: open access, traceability, and evolution. arXiv https://arxiv.org/abs/2004.00199.

123. Dozmorov, M. (2018). GitHub statistics as a measure of the impact of open-source bioinformatics software. Front. Bioeng. Biotechnol. *6*, 198.

124. Pimentel, J.F., Murta, L., Braganholo, V., and Freire, J. (2019). A large-scale study about quality and reproducibility of Jupyter notebooks. In 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), pp. 507–517. https://ieeexplore.ieee.org/document/8816763.

125. Boettiger, C. (2015). An introduction to docker for reproducible research. Oper. Syst. Rev. *49*, 71–79.

126. Grüning, B., Dale, R., Sjödin, A., Chapman, B.A., Rowe, J., Tomkins-Tinch, C.H., Valieris, R., and Köster, J.; Bioconda Team (2018). Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat. Methods *15*, 475–476.

127. ENCODE Project Consortium, Birney, E., Stamatoyannopoulos, J.A., Dutta, A., Guigó, R., Gingeras, T.R., Margulies, E.H., Weng, Z., Snyder, M., Dermitzakis, E.T., Thurman, R.E., et al. (2007). Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. Nature *447*, 799–816.

128. Hung, L.-H., Hu, J., Meiss, T., Ingersoll, A., Lloyd, W., Kristiyanto, D., Xiong, Y., Sobie, E., and Yeung, K.Y. (2019). Building containerized workflows using the BioDepot-workflow-builder. Cell Syst. *9*, 508–514.e3.

129. Moreews, F., Sallou, O., Ménager, H., Le Bras, Y., Monjeaud, C., Blanchet, C., and Collin, O. (2015). BioShaDock: a community driven bioinformatics shared docker-based tools registry. F1000Res. *4*, 1443.

130. Belmann, P., Dröge, J., Bremges, A., McHardy, A.C., Sczyrba, A., and Barton, M.D. (2015). Bioboxes: standardised containers for Interchangeable bioinformatics software. Gigascience *4*, 47.

131. da Veiga Leprevost, F., Grüning, B.A., Alves Aflitos, S., Röst, H.L., Uszkoreit, J., Barsnes, H., Vaudel, M., Moreno, P., Gatto, L., Weber, J., et al. (2017). BioContainers: an open-source and community-driven framework for software standardization. Bioinformatics *33*, 2580–2582.

132. Allamanis, M., and Sutton, C. (2013). Mining source code repositories at massive scale using language modeling. In Proceedings of the 10th Working Conference on Mining Software Repositories; MSR '13, T. Zimmerman, M. Di Penta, and S. Kim, eds. (IEEE Press), pp. 207–216.

133. Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., et al. (2004). Bioconductor: open software development for computational biology and bioinformatics. Genome Biol. *5*, R80.

134. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: machine learning in Python. J. Mach. Learn. Res. *12*, 2825–2830.

135. Tierney, N.J., and Ram, K. (2020). A realistic guide to making data available alongside code to improve reproducibility. arXiv https://arxiv.org/abs/2002.11626.

136. Cormier, M.J., Belyeu, J.R., Pedersen, B.S., Brown, J., Köster, J., and Quinlan, A.R. (2021). Go get data (GGD) is a framework that facilitates reproducible access to genomic data. Nat. Commun. *12*, 1–6.

137. Open Container Initiative https://www.opencontainers.org/.

138. Emsley, I., and De Roure, D. (2018). A framework for the preservation of a docker container. Int. J. Digit. Curation *12*, 125–135.

139. Rechert, K., Liebetraut, T., Wehrle, D., and Cochrane, E. (2016). Preserving containers – requirements and a todo-list. Digital libraries: knowledge, information, and data in an open access society, 225–230.

140. Yuen, D., Duncan, A., Liu, V., O'Connor, B., Patricia, J., oicr-vchung, Amstutz, P., and Badger, T.G. (2016). Ga4Gh/Dockstore: 1.0 (Zenodo).

141. Leisch, F. (2002). Sweave: dynamic generation of statistical reports using literate data analysis. In Compstat, P.D.W. Härdle and P.D.B. Rönz, eds. (Physica-Verlag HD), pp. 575–580.

142. Xie, Y. (2014). Knitr: a comprehensive tool for reproducible research in R. Implement Reprod. Res. *1*, 20.

143. RStudio (2015). RStudio: Integrated Development for Rl (RStudio).

144. Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., et al. (2016). Jupyter Notebooks—A Publishing Format for Reproducible Computational Workflows. Positioning and Power in Academic Publishing: Players, Agents and Agendas (IOS Press), pp. 87–90.

145. Shen, H. (2014). Interactive notebooks: sharing the code. Nature *515*, 151–152.

146. Zhang, S., Zhang, C., and Yang, Q. (2003). Data preparation for data mining. Appl. Artif. Intell. *17*, 375–381.

147. Rosenberg, D.M., and Horn, C.C. (2016). Neurophysiological analytics for all! Free open-source software tools for documenting, analyzing, visualizing, and sharing using electronic notebooks. J. Neurophysiol. *116*, 252–262.

148. Bussonnier, M., Forde, J., Freeman, J., Granger, B., Head, T., Holdgraf, C., et al.; Project Jupyter (2018). Binder 2.0-reproducible, interactive, sharable environments for science at scale. In Proceedings of the 17th Python in Science Conference, *113*, F. Akici, ed. (SciPy), p. 120.

149. Allaire, J.J., Xie, Y., Foundation, R., Wickham, H., RStudio, J. Stat. Softw., Vaidyanathan, R., Association for Computing Machinery, Boettiger, Elsevier, C., et al. (2016). Rticles: Article Formats for R Markdown.

150. Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Larochelle, H. (2020). Improving reproducibility in machine learning research (A report from the NeurIPS 2019 reproducibility program). arXiv.

151. Ćwiek-Kupczyńska, H., Filipiak, K., Markiewicz, A., Rocca-Serra, P., Gonzalez-Beltran, A.N., Sansone, S.-A., Millet, E.J., van Eeuwijk, F., Ławrynowicz, A., and Krajewski, P. (2020). Semantic concept schema of the linear mixed model of experimental observations. Sci. Data *7*, 70.

152. Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S.A., Konwinski, A., Murching, S., Nykodym, T., Ogilvie, P., Parkhe, M., et al. (2018). Accelerating the machine learning lifecycle with MLflow. IEEE Data Eng. Bull. *41*, 39–45.

153. Leipzig, J. (2016). A review of bioinformatic pipeline frameworks. Brief. Bioinform. *18*, 530–536. https://doi.org/10.1093/bib/bbw020.

154. Goecks, J., Nekrutenko, A., and Taylor, J.; Galaxy Team (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol. *11*, R86.

155. Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., and Mock, S. (2004). Kepler: an extensible system for design and execution of scientific workflows. In Proceedings of the 16th International Conference on Scientific and Statistical Database Management, 2004 (IEEE Computer Society), pp. 423–424.

156. Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Thiel, K., and Wiswedel, B. (2009). Knime - the Konstanz information miner: version 2.0 and beyond. SIGKDD Explor. Newsl. *11*, 26–31.

157. Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Li, P., and Oinn, T. (2006). Taverna: a tool for building and running workflows of services. Nucleic Acids Res. *34*, W729–W732.

158. Kaushik, G., Ivkovic, S., Simonovic, J., Tijanic, N., Davis-Dusenbery, B., and Kural, D. (2016). Rabix: an open-source workflow executor supporting recomputability and interoperability of workflow DescriptionS. Pac. Symp. Biocomput. *22*, 154–165.

159. Vivian, J., Rao, A.A., Nothaft, F.A., Ketchum, C., Armstrong, J., Novak, A., Pfeil, J., Narkizian, J., Deran, A.D., Musselman-Brown, A., et al. (2017). Toil enables reproducible, open source, big biomedical data analyses. Nat. Biotechnol. *35*, 314–316.

160. Robinson, M., Soiland-Reyes, S., Crusoe, M.R., and Goble, C. (2017). CWL viewer: the common workflow language viewer. In Bioinformatics Open Source Conference, N. Harris and H. Wiencko, eds. (BOSC), p. 2017.

161. Bandrowski, A.E., and Martone, M.E.R.R.I.Ds (2016). A simple step toward improving reproducibility through rigor and transparency of experimental methods. Neuron *90*, 434–436.

162. Pimentel, J.F., Freire, J., Murta, L., and Braganholo, V. (2019). A survey on collecting, managing, and analyzing provenance from scripts. ACM Comput. Surv. 1–38.

163. Lerner, B., and Boose, E. (2014). RDataTracker: collecting provenance in an interactive scripting environment. In 6th USENIX Workshop on the Theory and Practice of Provenance (TaPP).

164. Gehani, A., Kazmi, H., and Irshad, H. (2016). Scaling SPADE to "Big Provenance". In 8th USENIX Workshop on the Theory and Practice of Provenance (TaPP).

165. Angelino, E., Yamins, D., and Seltzer, M. (2010). StarFlow: a script-centric data analysis environment. In Provenance and Annotation of Data Processes, IPAW 2010, D.L. McGuiness, J.R. Michaelis, and L. Moreau, eds. (Springer), pp. 236–250.

166. McPhillips, T., Song, T., Kolisnik, T., Aulenbach, S., Belhajjame, K., Bocinsky, K., Cao, Y., Chirigati, F., Dey, S., Freire, J., et al. (2015). YesWorkflow: a user-oriented, language-independent tool for recovering workflow information from scripts. arXiv.

167. Freire, J. (2012). Making computations and publications reproducible with VisTrails. Comput. Sci. Eng. *14*, 18–25.

168. Garijo, D., Gil, Y., and Corcho, O. (2017). Abstract, link, publish, exploit: an end to end framework for workflow sharing. Future Gener. Comput. Syst. *75*, 271–283.

169. Nüst, D., Konkol, M., Pebesma, E., Kray, C., Schutzeichel, M., Przibytzin, H., and Lorenz, J. (2017). Opening the publication process with executable research compendia. D-Lib Mag. *23*. https://doi.org/10.1045/january2017-nuest.

170. Konkol, M., Kray, C., and Suleiman, J. (2019). Creating interactive scientific publications using bindings. Proc. ACM Hum.-Comput. Interact. *3*, 1–18.

171. Bechhofer, S., Buchan, I., De Roure, D., Missier, P., Ainsworth, J., Bhagat, J., Couch, P., Cruickshank, D., Delderfield, M., Dunlop, I., et al. (2013). Why linked data is not enough for scientists. Future Gener. Comput. Syst. *29*, 599–611.

172. Heery, R., and Patel, M. (2000). Application profiles: mixing and matching metadata schemas. Ariadne http://www.ariadne.ac.uk/issue/25/app-profiles/.

173. Duck, G., Nenadic, G., Brass, A., Robertson, D.L., and Stevens, R. (2014). Extracting patterns of database and software usage from the bioinformatics literature. Bioinformatics *30*, i601–i608.

174. Eales, J.M., Pinney, J.W., Stevens, R.D., and Robertson, D.L. (2008). Methodology capture: Discriminating between the "best" and the rest of community practice. BMC Bioinformatics *9*, 359.

175. Halioui, A., Valtchev, P., and Diallo, A.B. (2016). Towards an ontology-based recommender system for relevant bioinformatics workflows. bioRxiv. https://doi.org/10.1101/082776.

176. Sahoo, S.S., Valdez, J., Kim, M., Rueschman, M., and Redline, S. (2019). ProvCaRe: characterizing scientific reproducibility of biomedical research studies using semantic provenance metadata. Int. J. Med. Inform. *121*, 10–18.

177. Hrynaszkiewicz, I. (2020). Publishers' responsibilities in promoting data quality and reproducibility. Handb. Exp. Pharmacol. *257*, 319–348.

178. Nüst, D., Seibold, H., Eglen, S., Schulz-Vanheyden, L., Peer, L., and Spillner, J. (2021). Code Execution in Peer Review. Open Sci. Framework. https://doi.org/10.17605/OSF.IO/X32NC.

179. Evanko, D.. Guidelines for Algorithms and Software in Nature Methods. http://blogs.nature.com/methagora/2014/02/guidelines-for-algorithms-and-software-in-nature-methods.html.

180. Ince, D.C., Hatton, L., and Graham-Cumming, J. (2012). The Case for open computer programs. Nature *482*, 485–488.

181. Nüst, D., and Eglen, S.J. (2021). CODECHECK: an open science initiative for the independent execution of computations underlying research articles during peer review to improve reproducibility. F1000Res. *10*, 253.

182. Hucka, M., Bergmann, F.T., Chaouiya, C., Dräger, A., Hoops, S., Keating, S.M., König, M., Le Novère, N., Myers, C.J., Olivier, B.G., et al. (2019). The systems biology markup language (SBML): language specification for level 3 version 2 core release 2. J. Integr. Bioinform. *16*, 20190021.

183. Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M.I., Wimalaratne, S.M., et al. (2009). The systems biology graphical notation. Nat. Biotechnol. *27*, 735–741.

184. Demir, E., Cary, M.P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., Guanming, W., D'Eustachio, P., Schaefer, C., Luciano, J., et al. (2010). The BioPAX community standard for pathway data sharing. Nat. Biotechnol. *28*, 935–942.

185. The Gene Ontology Consortium (2019). The gene ontology resource: 20 Years and still GOing strong. Nucleic Acids Res. *47*, D330–D338.

186. Cerami, E.G., Gross, B.E., Demir, E., Rodchenkov, I., Babur, Ö., Anwar, N., Schultz, N., Bader, G.D., and Sander, C. (2011). Pathway commons, a web resource for biological pathway data. Nucleic Acids Res. *39*, 685–690.

187. Fabregat, A., Jupe, S., Matthews, L., Sidiropoulos, K., Gillespie, M., Garapati, P., Haw, R., Jassal, B., Korninger, F., May, B., et al. (2018). The reactome pathway Knowledgebase. Nucleic Acids Res. *46*, D649–D655.

188. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., and Morishima, K. (2017). KEGG: new perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res. *45*, D353–D361.

189. Perfetto, L., Briganti, L., Calderone, A., Perpetuini, A.C., Iannuccelli, M., Langone, F., Licata, L., Marinkovic, M., Mattioni, A., Pavlidou, T., et al. (2016). SIGNOR: a database of causal relationships between biological entities. Nucleic Acids Res. *44*, D548–D554.

190. Slenter, D.N., Kutmon, M., Hanspers, K., Riutta, A., Windsor, J., Nunes, N., Mélius, J., Cirillo, E., Coort, S.L., Digles, D., et al. (2018). WikiPathways: a multifaceted pathway database bridging metabolomics to other omics research. Nucleic Acids Res. *46*, D661–D667.

191. Hoyt, C.T., Domingo-Fernández, D., Aldisi, R., Xu, L., Kolpeja, K., Spalek, S., Wollert, E., Bachman, J., Gyori, B.M., Greene, P., et al. (2019). Re-curation and rational enrichment of knowledge graphs in biological Expression Language. Database *2019*, baz068.

192. Madan, S., Hodapp, S., Senger, P., Ansari, S., Szostak, J., Hoeng, J., Peitsch, M., and Fluck, J. (2016). The BEL information extraction workflow (BELIEF): evaluation in the BioCreative V BEL and IAT track. Database *2016*, baw136.

193. Allen, J.F., Swift, M., and De Beaumont, W. (2008). Deep semantic analysis of text. In Proceedings of the 2008 Conference on Semantics in Text Processing (Association for Computational Linguistics), pp. 343–354.

194. McDonald, D.D. (2000). Issues in the representation of real texts: the design of Krisp. In Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language, K. Iwanska and R. Shapiro, eds. (MIT Press), pp. 77–110.

195. Valenzuela-Escárcega, M.A., Babur, Ö., Hahn-Powell, G., Bell, D., Hicks, T., Noriega-Atala, E., Wang, X., Surdeanu, M., Demir, E., and Morrison, C.T. (2018). Large-scale Automated machine reading discovers new cancer-driving mechanisms. Database *2018*, bay098.

196. Gyori, B.M., Bachman, J.A., Subramanian, K., Muhlich, J.L., Galescu, L., and Sorger, P.K. (2017). From word models to executable models of signaling networks using automated assembly. Mol. Syst. Biol. *13*, 954.

197. Bachman, J.A., Gyori, B.M., and Sorger, P.K. (2018). FamPlex: a resource for entity recognition and relationship resolution of human protein families and complexes in biomedical text mining. BMC Bioinformatics *19*, 1–14.

198. Maciocci, G., Aufreiter, M., and Bentley, N. (2019). Introducing eLife's First Computationally Reproducible Article (eLife Labs). https://elifesciences.org/labs/ad58f08d/introducing-elife-s-first-computationally-reproducible-article.

199. Tsang, E., and Maciocci, G. (2020). Welcome to a new ERA of reproducible publishing, eLife Labs https://elifesciences.org/labs/dc5acbde/welcome-to-a-new-era-of-reproducible-publishing.

200. Guizzardi, G., Bentley, N., and Maciocci, G. (2021). Announcing the next phase of Executable Research Articles, eLife Labs https://elifesciences.org/labs/a04d2b80/announcing-the-next-phase-of-executable-research-articles.

201. Greenberg, J. (2005). Understanding metadata and metadata schemes. Cataloging classification Q. *40*, 17–36.

202. Prior, F., Smith, K., Sharma, A., Kirby, J., Tarbox, L., Clark, K., Bennett, W., Nolan, T., and Freymann, J. (2017). The public cancer radiology imaging collections of the cancer imaging archive. Sci. Data *4*, 170124.

203. Pérez, W., Tello, A., Saquicela, V., Vidal, M., and La Cruz, A. (2015). An automatic method for the enrichment of DICOM metadata using biomedical ontologies. In 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2551–2554.

204. Bourne, P.E. (2015). DOIs for DICOM raw images: enabling science reproducibility. Radiology *275*, 3–4.

205. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R.; 1000 Genome Project Data Processing Subgroup (2009). The sequence alignment/map format and SAMtools. Bioinformatics *25*, 2078–2079.

206. Queralt-Rosinach, N., Piñero, J., Bravo, À., Sanz, F., and Furlong, L.I. (2016). DisGeNET-RDF: Harnessing the innovative power of the semantic web to explore the genetic basis of diseases. Bioinformatics *32*, 2236–2238.

207. Janowicz, K., and Hitzler, P. (2016). Geospatial Semantic Web. In International Encyclopedia of Geography: People, the Earth, Environment and Technology, *284*, D. Richardson, N. Castree, M.F. Goodchild, A. Kobayashi, W. Liu, and R.A. Marston, eds. (John Wiley & Sons, Ltd), pp. 1–6.

208. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: A nucleus for a web of open data. In The Semantic Web, K. Aberer, K.S. Choi, N. Noy, D. Allenmang, K.I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, and R. Mizoguchi, eds. (Springer), pp. 722–735.

209. Dumontier, M., Callahan, A., Cruz-Toledo, J., Ansell, P., Emonet, V., Belleau, F., et al. (2014). Bio2RDF release 3: a larger connected network of linked data for the life sciences. In Proceedings of the 2014 International Conference on Posters & Demonstrations Track, *1272*, M. Horridge, M. Rospocher, and J. van Ossenbruggen, eds. (Association for Computing Machinery), pp. 401–404.

210. Kulmanov, M., Smaili, F.Z., Gao, X., and Hoehndorf, R. (2020). Machine learning with biomedical ontologies. bioRxiv. https://doi.org/10.1101/2020.05.07.082164.

211. Ali, M., Jabeen, H., Hoyt, C.T., and Lehman, J. (2020). The KEEN universe: an ecosystem for knowledge graph embeddings with a focus on reproducibility and transferability. arXiv https://arxiv.org/abs/2001.10560.

212. Stein, L.D. (2010). The case for cloud computing in genome Informatics. Genome Biol. *11*, 207.

213. De Roure, D., Manuel, J., Hettne, K., Belhajjame, K., Palma, R., Klyne, G., et al. (2011). Towards the Preservation of Scientific Workflows. In Proceedings of the 8th International Conference on Preservation of Digital Objects (iPRES 2011), J. Borbinha, ed. (Association for Computing Machinery).

214. Courtot, M., Cherubin, L., Faulconbridge, A., Vaughan, D., Green, M., Richardson, D., Harrison, P., Whetzel, P.L., Parkinson, H., and Burdett, T. (2019). BioSamples database: an updated sample metadata hub. Nucleic Acids Res. *47*, D1172–D1178.

215. Aranguren, M.E., and Wilkinson, M.D. (2015). Enhanced reproducibility of SADI web service workflows with Galaxy and Docker. Gigascience *4*, 59.

216. Frey, J.G., and Bird, C.L. (2013). Cheminformatics and the semantic web: adding value with linked data and enhanced provenance. Wiley Interdiscip. Rev. Comput. Mol. Sci. *3*, 465–481.

217. Simonyan, V., Goecks, J., and Mazumder, R. (2017). Biocompute objects-A step towards evaluation and validation of biomedical scientific computations. PDA J. Pharm. Sci. Technol. *71*, 136–146.

218. Chirigati, F., Rampin, R., Shasha, D., and Freire, J. (2016). ReproZip: computational reproducibility with ease. In Proceedings of the 2016 International Conference on Management of Data; SIGMOD '16, F. Özcan, G. Koutrika, and S. Madden, eds. (Association for Computing Machinery), pp. 2085–2088.

219. Love, M.I., Soneson, C., Hickey, P.F., Johnson, L.K., Pierce, N.T., Shepherd, L., Morgan, M., and Patro, R. Tximeta (2020). Reference sequence checksums for provenance identification in RNA-seq. PLoS Comput. Biol. *16*, e1007664.

220. Greenberg, J. (2017). Big metadata, smart metadata, and metadata capital: toward greater synergy between data science and metadata. J. Data Inf. Sci. *2*, 193.

221. Wang, H., and Webster, K. (2019). Artificial intelligence for data discovery and reuse demands healthy data ecosystem and community efforts. In Proceedings of the Conference on Artificial Intelligence for Data Discovery and Reuse, H. Wang and K. Webster, eds. (National Science Foundation).

222. Murillo, A.P. (2014). Examining data sharing and data reuse in the DataONE environment. Proc. Am. Soc. Inf. Sci. Technol. *51*, 1–5.

223. Bernstein, M.N., Doan, A., and Dewey, C.N. (2017). MetaSRA: Normalized human sample-specific metadata for the sequence read archive. Bioinformatics *33*, 2914–2923.

224. DUO: the Data Use Ontology (Github).

225. LeVeque, R.J., Mitchell, I.M., and Stodden, V. (2012). Reproducible research for scientific computing: tools and strategies for changing the culture. Comput. Sci. Eng. *14*, 13.

226. Arabas, S., Bareford, M.R., de Silva, L.R., Gent, I.P., Gorman, B.M., Hajiarabderkani, M., Henderson, T., Hutton, L., Konovalov, A., Kotthoff, L., et al. (2014). Case studies and challenges in reproducibility in the computational sciences. arXiv.

227. Rosenberg, D.E., Jones, A.S., Filion, Y., Teasley, R., Sandoval-Solis, S., Stagge, J.H., Abdallah, A., Castronova, A., Ostfeld, A., and Watkins, D., Jr. (2021). Reproducible results policy. J. Water Resour. Plan. Manag. *147*, 01620001.

228. Nüst, D., Lohoff, L., Einfeldt, L., Gavish, N., Götza, M., Jaswal, S., Khalid, S., Meierkort, L., Mohr, M., and Rendel, C. (2019). Guerrilla Badges for Reproducible Geospatial Data Science. Earth ArXiv. https://doi.org/10.31223/osf.io/xtsqh.

229. NICTA Optimisation Research Group (2014). NICTA-ORG/MLG seminar: C. Titus Brown - openness and reproducibility in computational science. https://www.youtube.com/watch?v=12hpAYr5ls0.

230. Schimanski, L.A., and Alperin, J.P. (2018). The evaluation of scholarship in academic promotion and tenure processes: past, present, and future. F1000Res. *7*, 1605.

231. Katz, D.S., Chue Hong, N.P., Clark, T., Muench, A., Stall, S., Bouquin, D., Cannon, M., Edmunds, S., Faez, T., Feeney, P., et al. (2020). Recognizing the value of software: a software citation guide. F1000Res. *9*, 1257.

232. Smith, A.M., Niemeyer, K.E., Katz, D.S., Barba, L.A., Githinji, G., Gymrek, M., Huff, K.D., Madan, C.R., Mayes, A.C., Moerman, K.M., et al. (2018). Journal of open source software (JOSS): design and first-year review. Peerj Comput. Sci. *4*, e147.

233. Clarke, D.J.B., Wang, L., Jones, A., Wojciechowicz, M.L., Torre, D., Jagodnik, K.M., Jenkins, S.L., McQuilton, P., Flamholz, Z., Silverstein, M.C., et al. (2019). FAIRshake: toolkit to evaluate the FAIRness of research digital resources. Cell Syst. *9*, 417–421.

234. Simera, I., Moher, D., Hirst, A., Hoey, J., Schulz, K.F., and Altman, D.G. (2010). Transparent and accurate reporting increases reliability, utility, and impact of Your research: reporting guidelines and the EQUATOR network. BMC Med. *8*, 24.

235. Schulz, K.F., Altman, D.G., and Moher, D.; CONSORT Group (2010). CONSORT 2010 statement: updated guidelines for reporting parallel group randomised trials. BMC Med. *8*, 18.

236. Himmelstein, D.S., Rubinetti, V., Slochower, D.R., Hu, D., Malladi, V.S., Greene, C.S., and Gitter, A. (2019). Open collaborative writing with Manubot. PLoS Comput. Biol. *15*, e1007128.

237. Anzt, H., Bach, F., Druskat, S., Löffler, F., Loewe, A., Renard, B.Y., Seemann, G., Struck, A., Achhammer, E., Aggarwal, P., et al. (2021). An environment for sustainable research software in Germany and beyond: current state, open challenges, and call for action. F1000Res. *9*, 295.

238. Landau, W. (2021). The targets R package: a dynamic make-like function-oriented pipeline toolkit for reproducibility and high-performance computing. J. Open Source Softw. *6*, 2959.

239. Dippo, C.S., and Sundgren, B. (2000). The Rold of Metadata in Statistics. U.S. Bureau of Labor Statistics https://www.bls.gov/osmr/research-papers/2000/st000040.htm.

240. Mangul, S., Martin, L.S., Hill, B.L., Lam, A.K.-M., Distler, M.G., Zelikovsky, A., Eskin, E., and Flint, J. (2019). Systematic benchmarking of omics computational tools. Nat. Commun. *10*, 1393.