

The Role of Metadata in Reproducible Computational Research

Jeremy Leipzig₁; Daniel Nüst₂; Charles Tapley Hoyt₃; Stian Soiland-Reyes_{4,5}; Karthik Ram₆; Jane Greenberg₁

₁ Metadata Research Center, Drexel University, College of Computing and Informatics, Philadelphia PA, USA

₂ Institute for Geoinformatics, University of Münster, Münster, Germany

₃ Envada Therapeutics, Bonn, Germany

₄ eScience Lab, Department of Computer Science, The University of Manchester, Manchester, UK

₅ INDE lab, Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands

₆ Berkeley Institute for Data Science, University of California, Berkeley, USA

Abstract

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, RCR has the capacity to significantly accelerate evaluation and reuse. This potential and wide-support for the FAIR principles have motivated interest in metadata standards supporting RCR. Metadata provides context and provenance to raw data and methods and is essential to both discovery and validation. Despite this shared connection with scientific data, few studies have explicitly described the relationship between metadata and RCR. This article employs a functional content analysis to identify metadata standards that support RCR functions across an analytic stack consisting of input data, tools, notebooks, pipelines, and publications. Our article provides background context, explores gaps, and discovers component trends of embeddedness and methodology weight from which we derive recommendations for future work.

Keywords: reproducible research, reproducible computational research, RCR, reproducibility, replicability, metadata, provenance, workflows, pipelines, ontologies, notebooks, containers, software dependencies, semantic, FAIR

Contents

Introduction	3
Reproducible Computational Research	3
Reproducibility Crisis	4
Big Data, Big Science, and Open Data	6
Metadata	8
Goals and Methods	9
The RCR metadata stack	10
Synthesis Review	10
1. Input	12
Examples	13
DICOM - An embedded file header	13
EML - Flexible user-centric data documentation	13
MIAME - A submission-centric minimal standard	14
Future directions - encoding, findability, granularity	16
2. Tools	16
Examples	17
CRAN, EDAM, & CodeMeta - Tool description and citation	17
Dependency and package management metadata	19
Fledgling standards for containers	19
Future directions	20
Automated repository metadata	20
Data as a dependency	20
3. Statistical reports & Notebooks	21
Examples	22
RMarkdown headers	22
Statistical and Machine Learning Metadata Standards	23
Future directions - parameter tracking	24
4. Pipelines	24
Examples	26
CWL - A configuration-based framework for interoperability	26
Future directions	28
Interoperable script and workflow provenance	28
Packaging and binding building blocks	28
5. Publication	31
Examples	32
Formalization of the Results of Biological Discovery	32
Future directions - reproducible articles	33
Discussion	33
Embeddedness vs connectedness	34
Methodology weight and standardization	36
Sphere of Influence	36
Metadata capital and reuse	37
Recommendations & Future Work	38
Acknowledgements	40
References	40

Introduction

Digital technology and computing have transformed the scientific enterprise. As evidence, many scientific workflows 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 has emerged as a crucial component, supporting these advances, with standards supporting the wide array of tasks and functions that form the research life-cycle. Reflective of change, there have been both many case studies on reproducibility, and many on metadata standards, although few have attempted to systematically study their relationship. Our aim in this work is to review metadata developments that are directly applicable to RCR, identify gaps, and recommend further steps involving metadata toward building a more robust RCR environment. To lay the groundwork for these recommendations, we first review the RCR and metadata, examine how they relate across different stages of an analysis, and discuss what common trends emerge from this approach.

Reproducible Computational Research

Reproducible Research is an umbrella term that encompasses many forms of scientific quality - from generalizability of underlying scientific truth, exact replication of an experiment with or without communicating intent, to the open sharing of analysis for reuse. Specific to computational facets of scientific research, *Reproducible Computational Research* (RCR)⁵ 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, to open code and statistical analysis, and finally publication. Reproducible research points to several underlying concepts of scientific validity – terms that should be unpacked to be understood. Stodden et al.⁶ devised a five-level hierarchy of research, classifying it as – reviewable, replicable, confirmable, auditable, and open or reproducible. Whitaker⁷ describes an analysis as "reproducible" in the narrow sense that a user can produce identical results provided the data and code from the original, and "generalisable" if it produces similar results when both data is swapped out for similar data ("replicability"), and if underlying code is swapped out with comparable

replacements ("robustness") (Figure 1).

		Data	
		Same	Different
Analysis	Same	Reproducible	Replicable
	Different	Robust	Generalisable

Figure 1: Whitaker's matrix of reproducibility ⁸

While these terms may confuse those new to reproducibility, a review by Barba disentangles the terminology while providing a historical context of the field ⁹. A wider perspective places reproducibility as a first-order benefit of applying FAIR principles: Findability, Accessibility, Interoperability, and Reusability. In the next sections, we will engage reproducibility in the general sense and will use "narrow-sense" to refer to the same data, same code condition.

Reproducibility Crisis

In recent years, the scientific community has been grappling with the problem of irreproducibility in research. Two events in the life sciences stand out 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 ¹⁰, and a review conducted by scientists at Amgen who could replicate the results of only 6 out of 53 cancer studies ¹¹. These events involved different aspects - poor data structures and missing protocols, respectively, and related studies ¹² have identified recurring reproducibility problems due to a lack of detailed methods, missing controls, and other failures in protocol. An inadequate understanding of statistics, which may include the application of inappropriate statistical tests and misinterpretation or abuse of statistical tests, is believed to play a recurring role in irreproducibility ¹³. It bears speculation whether the risk of these types of incidents is more likely to occur in novel statistical approaches than in conventional ones. Subsequent surveys of researchers ¹⁴ have identified selective reporting, while theory papers ¹⁵ have emphasized the insidious combination of underpowered designs and publication bias, essentially a multiple testing problem on a global scale. It is our contention that RCR metadata

has a role to play in addressing all of these issues and to shift the narrative from a crisis to opportunities ¹⁶.

In the wake of this newfound interest in reproducibility, both the variety and volume of related case studies has exploded since 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. Additionally, low-level reproductions, in which code is executed, replications in which new data is 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.

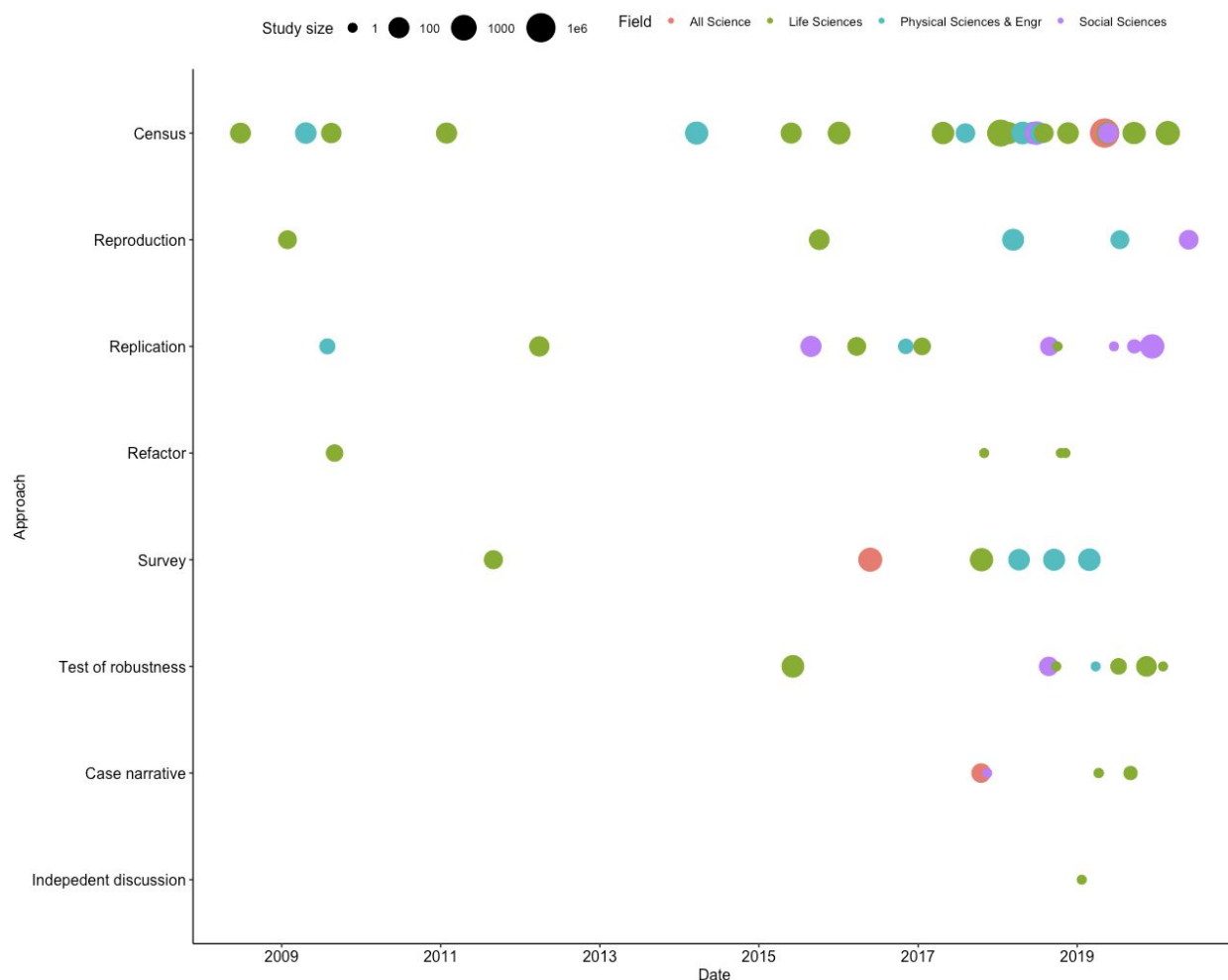


Figure 2: Case studies in reproducible research ¹⁷. The term "case studies" is used here in a general sense to describe any study of reproducibility. A reproduction is an attempt to arrive at comparable results with identical data using computational methods described in a paper. A refactor involves refactoring existing code into frameworks and other reproducibility 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 data set to study their effect on results, either as a follow-up to an existing study or as a "bake-off". 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 utilizes 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 ¹⁸ but the scale of scientific endeavor and the level of data and method reuse suggest replication failures may actually 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 data sets are generated and analyzed by multi-institutional collaborative research projects. Metadata aspects of big data have been quantitatively studied with regard to reuse ^{19,20}, but not reproducibility, despite some evidence big data may play a role in spurious results associated with reporting bias ²¹. 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 ^{22,23}. RCR 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 ²⁴.

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 ²⁵⁻²⁸.

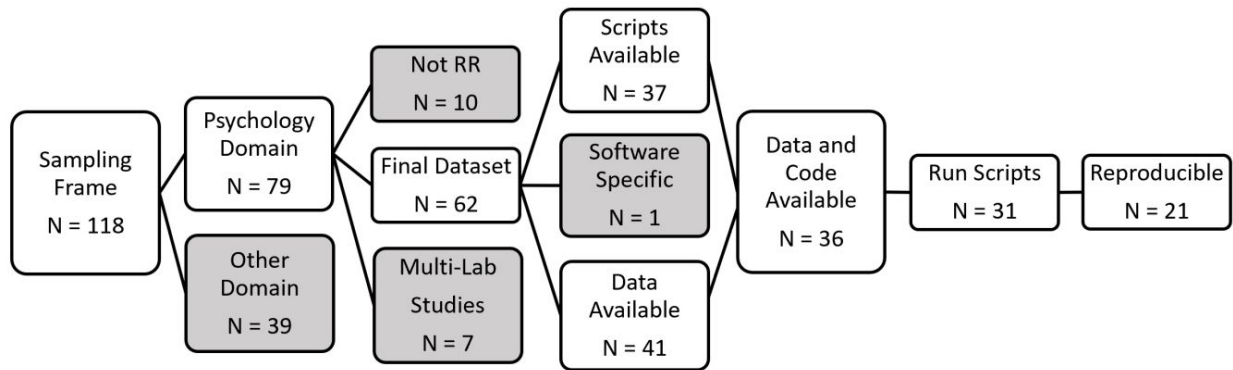


Figure 3. Censuses like this one by Obels et al. measure reproducibility with respect to data and code availability, in this case over a corpus of 118 studies ²⁵.

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 ²⁹. Sandve et al. ³⁰ 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 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 ³¹, but also steps along the entire scientific process including data collection and selection strategies, data processing provenance, 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 ³² – the ability to execute a packaged analysis with little effort from 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 ³³. 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 last twenty-five years, metadata has gained acceptance as a key component of research infrastructure design. This trend is defined by numerous initiatives supporting the development and sustainability of hundreds of metadata standards, each with varying characteristics ^{34,35}. Across these developments, there is a general high-level consensus regarding the following three types of metadata standards ^{36,37}:

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 affiliate with resource use. Administrative metadata includes technical, preservation, and rights metadata.
3. *Structural metadata*, supporting the linking among the component parts of a resource, so it can be fully understood.

There is also general agreement that metadata is 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 ³⁸), "terminology artifacts or semantics" (formal taxonomies or ontologies to disambiguate concepts e.g. Gene Ontology ³⁹), "models and formats" (e.g. FASTA ⁴⁰), "metrics" (e.g. FAIRMetrics ⁴¹) and "identifier schemata" (e.g. DOI ⁴²) ⁴³. (See Table Y).

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

Metadata is 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 RCR metadata broadly as **any structured data that aids reproducibility and that can conform to a standard**. While this definition may seem liberal, it is our contention that metadata is the "glue" of RCR, 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 RCR, together with the fact that there are few

studies targeting metadata specifically along 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 RCR, identify gaps, discuss common threads among these efforts, and recommend next steps toward building a more robust RCR environment further work.



Figure 4: Terms enriched in the review corpus

Our method is framed as a state of the art review based on literature (Figure 4) and ongoing software development in the scientific community. Review steps included: **1)** defining key components of the RCR analytic stack, and function 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 RCR functions, and **4)** designing the RCR metadata hierarchy. Our approach was informed, in part, by the Qin LIGO case study ⁴⁴, catalogs of metadata standards such as FAIRSharing, and comprehensive projects to bind semantic science such as Research Objects ⁴⁵. 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 reveals most general terms.

The RCR metadata stack

To define the key aspects of RCR, 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. publication. These levels correspond loosely to the data science (Data understand, prep, modeling, evaluation, deployment), 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) ⁴⁶ and software development communities (Plan, Collect, Quality Control, Document, Preserve, Use). However, unlike the steps in lifecycle we do not emphasize a strong temporal order to these layers, but instead consider them simply interactive components of any scientific output.

Synthesis Review

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. Further blurring the lines, some standards could arguably be classified into two to more areas. For example, because manuscripts are the "final" product of scientific research, standards that attempt to bind all products of a scientific analysis could logically be associated with publications, but binding solutions (e.g. RO-Crate) are ostensibly associated with pipelines. Similarly, some solutions are flexible enough to be repurposed - script-based pipelines can be extended to notebooks and more flexible input metadata such (e.g. EML) can be used for publication-level annotation of results. In these cases, we tried to assign standards to their original intent.

The synthesis below first presents a high-level summary table, followed by a more detailed description of each of five levels, specific examples, and a forecast of future directions.

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 , Biotooolsxsd, 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,	Bibliographic, Scientific field,	BEL , Dublin Core, JATS, ONIX, MeSH,	NeuroLibre, JOSS,

	attribution	Scientific approach (e.g. "GWAS")	LCSH, MP, Open PHACTS, SWAN, SPAR, PWO, PAV	ReScience, Manubot
--	-------------	-----------------------------------	---	--------------------

Table 2: Metadata standards including: MIAME³⁸, EML⁴⁷, DICOM⁴⁸, GBIF⁴⁹, CIF⁵⁰, ThermoML⁵¹, CellML⁵², DATS⁵³, FAANG⁵⁴, ISO/TC 276⁵⁵, GO³⁹, Biotoolsxsd⁵⁶, meta.yaml⁵⁷, DOAP⁵⁸, ontosoft⁵⁹, EDAM⁶⁰, SWO⁶¹, OBCS⁶², STATO⁶³, SDMX⁶⁴, DDI⁶⁵), MEX⁶⁶, MLSchema⁶⁷, CWL⁶⁸, WICUS⁶⁹, OPM⁷⁰, PROV-O⁷¹, CWLProv⁷², ProvOne⁷³, PAV⁷⁴, BagIt⁷⁵, RO⁴⁵, RO-Crate⁷⁶, BCO⁷⁷, Dublin Core⁷⁸, JATS⁷⁹, ONIX⁸⁰, MeSH⁸¹, LCSH⁸², MP⁸³, Open PHACTS⁸⁴, BEL⁸⁵, SWAN⁸⁶, SPAR⁸⁷, PWO⁸⁸. Standards in **bold** are featured within this article. Examples of all standards can be found at <https://github.com/leipzig/metadata-in-rcr>

1. Input

Input refers to raw data from wet lab, field, instrumentation, or public repositories, intermediate processed files, and results from manuscripts. Compared to other layers of the analytic stack, input data garners 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. Metadata about raw data, such as variable labels and data definition tables, is among the oldest forms of metadata, but it plays important, and often unanticipated, roles in RCR. Most instrumentation, field measurements, and wet lab protocols are communicated through metadata and are useful for detecting anomalies such as batch effects and sample mix-ups.

While metadata is 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 with some level of inconvenience 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 Foundry⁸⁹ and NCBI BioPortal serve as catalogues of life science ontologies. The usage of these ontologies appear to follow a steep Pareto distribution, with "Gene Ontology" garnering more than 20,000 term mentions in PubMed, the vast majority of NCBO's 843 ontologies have never been cited or mentioned.

Examples

In addition to being the oldest, and arguably most visible of RCR metadata standards, input metadata standards serve as a watershed for downstream reproducibility. In order to understand what input means for RCR, we will 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 1985⁹⁰. 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 scheme⁹¹. 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 has suffered from "tag entropy" in the form of missing, incorrectly filled, non-standard, or misused tags by technicians manually entering in metadata⁹². This can pose problems both for clinical workflows as well as efforts to aggregate imaging data for the purposes of data mining and machine learning. The data structures imposed by the DICOM header format also hamper the level of granularity required to embed advanced annotations supporting image segmentation and quantitative analysis. This has made it necessary for programs such as 3DSlicer⁹³ and its associated plugins, such as dcqmi⁹⁴ 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 data⁴⁷. 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 5).

```

<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>

```

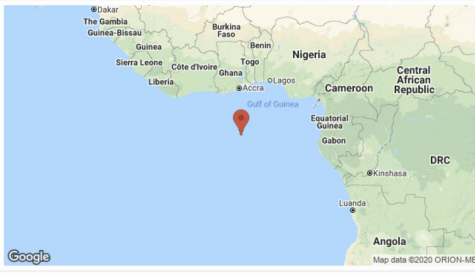
Geographic Region									
Geographic Description	Global Oceans								
Bounding Coordinates	<table border="1"> <tbody> <tr> <td>North</td> <td>90 degrees</td> </tr> <tr> <td>South</td> <td>-90 degrees</td> </tr> <tr> <td>East</td> <td>180 degrees</td> </tr> <tr> <td>West</td> <td>-180 degrees</td> </tr> </tbody> </table>	North	90 degrees	South	-90 degrees	East	180 degrees	West	-180 degrees
North	90 degrees								
South	-90 degrees								
East	180 degrees								
West	-180 degrees								
									
Temporal Coverage									
Date Range	<table border="1"> <tbody> <tr> <td>Begin</td> <td>2008</td> </tr> <tr> <td>End</td> <td>2013</td> </tr> </tbody> </table>	Begin	2008	End	2013				
Begin	2008								
End	2013								

Figure 5: Geographic and temporal EML metadata and the associated display on Knowledge Network for Biocomplexity (KNB) from Halpern et al.⁹⁵

MIAME - A submission-centric minimal standard

Minimum Information About a Microarray Experiment (MIAME)³⁸ 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 RCR: to support structured and computable experimental features ⁹⁶.

```
<?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>
Typical commercial baker's yeast used in Japan
    </Characteristics>
    <Treatment-Protocol>
```

Figure 6: An example of MIAME in MINiML format

(https://www.ncbi.nlm.nih.gov/geo/info/MINiML_Affy_example.txt)

MIAME (Figure 6) 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 ⁹⁷. However, it should be noted that among MIAME and other ISA-based standards that have followed suit ⁹⁸, none offer a controlled vocabulary for describing downstream computational workflows aside from slots to name the normalization procedure applied to what are essentially unitless intensity values.

Future directions - encoding, findability, granularity

Metadata for input is developing along descriptive, administrative, and structural axes. Scientific computing has continuously and selectively adopted a number of 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 ⁹⁹.

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) ¹⁰⁰. While not fully realized, this vision continues to evolve and encompass scientific data. Schema.org, a consortium of e-commerce companies developing tags for markup and discovery, such as those respected by Google Dataset Search ¹⁰¹, has coalesced a stable set of tags that is expanding into scientific domains. The potential for semantic markup of documents and scientific terms inline has huge potential for findability. In terms of reproducibility, Schema.org can be used to identify and distinguish in inputs and outputs of analyses in a disambiguated and machine-readable fashion. Another metadata-centric effort toward improving the findability of datasets is DATS ⁵³, a Schema.org-compatible tag-suite to describe fundamental metadata for datasets akin to that used for journal articles.

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 ¹⁰². Allotrope uses the W3C Shapes Constraint Language (SHACL) to describe which RDF relationships are valid to describe lab operations.

2. 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 homogenous 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 ¹⁰³.

Examples

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

CRAN, EDAM, & 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 a number of packaging manifests with excellent metadata standards exist for several languages, such as R's Comprehensive R Archive Network (CRAN) DESCRIPTION files (Figure 7).

```
Package: DESeq2
Type: Package
Title: Differential gene expression analysis based on the negative
       binomial distribution
Version: 1.27.31
Authors@R: c(
  person("Michael", "Love", email="michaelisaiahlove@gmail.com", role = c("aut", "cre")),
  person("Constantin", "Ahlmann-Eltze", role = c("ctb")),
  person("Simon", "Anders", role = c("aut", "ctb")),
  person("Wolfgang", "Huber", role = c("aut", "ctb")))
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)
LinkingTo: Rcpp, RcppArmadillo
URL: https://github.com/mikelove/DESeq2
biocViews: Sequencing, RNASeq, ChIPSeq, GeneExpression, Transcription,
           Normalization, DifferentialExpression, Bayesian, Regression,
           PrincipalComponent, Clustering, ImmunoOncology
RoxygenNote: 6.1.1
Encoding: UTF-8
```

Figure 7. An R package DESCRIPTION file from DESeq2 ¹⁰⁴

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 significant effects of an analysis ¹⁰⁵. 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 ⁶⁰. It has been used extensively in tool recommenders ¹⁰⁶, tool registries ¹⁰⁷, and within pipeline frameworks and workflow languages ^{108,109}. 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 ¹¹⁰. For better or worse, this reduces the need for workbench developers to manually annotate tools with ontologies, replacing them with a machine learning black box.

CodeMeta ¹¹¹ 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 (Figure 8).

```
{
  "@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. Th",
  "name": "baydem: Bayesian Tools for Reconstructing Past and Present\n  Demography",
  "license": "https://spdx.org/licenses/MIT",
  "version": "0.1.0",
  "programmingLanguage": {
    "@type": "ComputerLanguage",
    "name": "R",
    "version": "3.6.3",
    "url": "https://r-project.org"
  },
  "runtimePlatform": "R version 3.6.3 (2020-02-29)",
}
```

Figure 8: A snippet of CodeMeta JSON file from Price et al. ¹¹² using Schema.org contextual tags

Considerable strides have been made in improving software citation standards ¹¹³, which should improve the provenance of methods sections that cite those tools that do not already have accompanying manuscripts. Related to this in terms of code attribution is the compelling application of large-scale data mining and computer language processing in code repositories such as Github is the generation of dependency networks ¹¹⁴, measures of impact ¹¹⁵, and reproducibility censuses ¹¹⁶.

Dependency and package management metadata

Compiled software often depends on libraries that are shared by many programs on an operating system. Conflicts between versions of these libraries, and software that demands obscure or outdated versions of these libraries, is 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 largely considered the primary stumbling block to reproducible research ¹¹⁷. 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 a number of competing and evolving tools, such as pip, Pipenv, Conda, and Poetry in the Python community, and even different conceptual approaches, such as the CRAN time machine. In R, the default installation method does not support installing a specific version, but instead, all packages are rigorously tested to not break dependent packages for any given point in time. To free a specific state, users define the date from when packages should be installed from MRAN/CRAN time machine. In recent years, a growing number of scientific software projects utilize combinations of Python and compiled software, which is outside the scope of pip. The Conda project (<https://conda.io>) was developed to provide a universal solution for software dependencies written in any language. Both compiled executables and script dependencies, even across programming languages and the versions of the languages themselves can be interspersed with versioned Conda requirements specifications files, which serve as a single configuration for an entire project. The elegance of providing a single requirements file has contributed to Conda's rapid adoption for domain-specific library collections such as Bioconda ¹¹⁸, which are maintained in "channels" which 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 operating system. The ENCODE project ¹¹⁹ 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 cloud-based workbenches and other scientific environments where dependencies and permissions become unruly. A number of papers have demonstrated the usefulness of Docker for reproducible workflows ^{117,120} and as a central unit of tool distribution ^{121,122}.

There are a number of projects that now allow Conda programs to be automatically Dockerized, notably every BioConda package gets a corresponding BioContainer ¹²³ image built for Docker and Singularity. Because Dockerfiles are similar to shell scripts, Docker metadata is 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

(Figure 9). The Open Container Initiative's Image Format Specification (<https://github.com/opencontainers/image-spec/>) defines a number of 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. There are also licensing and security concerns when using and extending a *base image* with binaries of unknown origins. Unless storage space is an issue, storing both a Dockerfile and the Docker image, which contains the same metadata in a JSON format, can be an option to increase the chances of preservation.

```

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)"

```

Figure 9: 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

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 ¹²⁴.

Data as a dependency

Data libraries, which pair data sources with common programmatic methods for querying them are very popular in centralized open source repositories such as Bioconductor ¹²⁵, and scikit-learn ¹²⁶, despite often being large downloads. Tierney and Ram provide a best practices guide to the organization and necessary metadata for data libraries and independent data sets ¹²⁷. 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 includes a limited number of formats, but ideally, data should be distributed in packages bound to a variety of tested formatters. One solution, Gogetdata (<https://gogetdata.github.io/>), 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 ¹²⁸ is working to standardize "filesystem bundles" - the collection of files in a container and their metadata. In particular, container metadata is 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.

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 container store. An example of such a database is the GA4GH Dockstore ¹²⁹, 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.

3. Statistical reports & Notebooks

Statistical reports and notebooks serve as an annotated session of an analysis. Though they typically use input data that has been processed by scripts and workflows (layer 4 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 actually introduce additional challenges - one reason being the metadata supporting them is surprisingly sparse.

Statistical reports which utilize *literate programming*, combining statistical code with descriptive text, markup, and visualizations have been a standard for statistical communication since the advent of Sweave ¹³⁰. Sweave allowed R and LaTeX markup to be mixed in chunks, allowing adjacent contextual description of statistical code to serve as guideposts for anyone reading a Sweave report, typically rendered as PDF. An evolution of Sweave, knitr ¹³¹, extended choices of both markup (allowing Markdown) and output (html) while enabling tighter integration with integrated development environments such as RStudio ¹³². A related project which started in the Python ecosystem but now supports a number of kernels, Jupyter ¹³³, combined the concept of literate programming with a REPL (read-eval-print loop) in a web-based interactive session in which each block of code is kept stateful and can be re-evaluated. 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. Data-set specific Jupyter notebooks "spawned" for thousands of individuals on a temporary basis have been enabled as companions for Nature

articles ¹³⁴ 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 (<https://www.kaggle.com/>) 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 ¹³⁵. Sharing capabilities in existing open source notebook platforms are at a nascent stage, but this presents significant 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 ^{136,137}.

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, therefore they represent an important area for reproducibility.

Examples

RMarkdown headers

As we mentioned, statistical reports and notebooks have not yet received a great deal of attention with respect to reproducibility through structured metadata. R Markdown based reports, such as those processed by knitr, do have a YAML-based header (Figure 10). These are used for a wide variety of technical parameters for controlling display options, for providing metadata on authors, e.g., when used for scientific publications with the *rticles* package ¹³⁸, or for parameterizing the included workflow (https://rmarkdown.rstudio.com/developer_parameterized_reports.html%23parameter_types%2E). However, no schema or standards exist for their validation.

```
---
title: "A title for the analysis"
output:
  html_document:
    theme: lumen
    toc: true
    toc_float:
      collapsed: false
    code_folding: show
---
```


Figure 10: A YAML-based RMarkdown header from https://github.com/jmonlong/MonBUG18_RMarkdown

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 standards¹³⁹. 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 non-experts suggest it is necessary to begin cataloguing 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 11). 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.

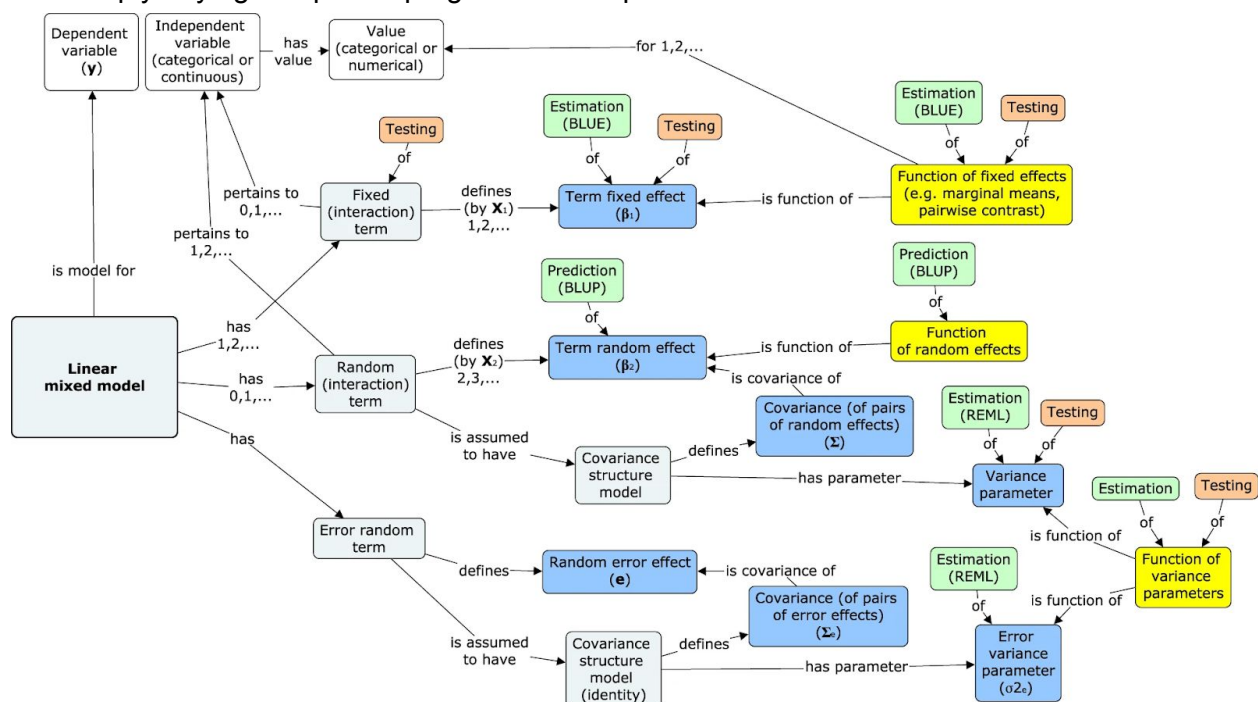


Figure 11: Concepts pertaining to a linear mixed model used by STATO¹⁴⁰

One attempt to ontologize the machine learning workflows. 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 ¹⁴¹ 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 MLFlow uses exposes variables which are explored and tuned by end-users (Figure 12).

```
name: HyperparameterSearch

conda_env: conda.yaml

entry_points:
  # train Keras DL model
  train:
    parameters:
      training_data: {type: string, default: "../sklearn_elasticnet_wine/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}"
```

Figure 12: MLflow snippet showing exposed hyperparameters

4. 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," a term which sometimes also 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 give 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. Leipzig¹⁴² 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 cluster computing frameworks such as Apache Spark.

Configuration based framework abstract pipelines into configuration files, typically XML or JSON which contain little or no code. Programmatic logic, for instance, determining which tasks can be run in parallel, which tasks are not relevant to a particular data set and can be skipped, must be represented using predefined configuration properties or wrapped in helper tasks. Workbenches such as Galaxy¹⁴³, Kepler¹⁴⁴, KNIME¹⁴⁵, Taverna¹⁴⁶, 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." Configuration-based pipelines offer the greatest potential for the distribution of reproducible workflows to others, and the easiest integration of metadata concepts discussed in this paper but can be onerous to build for one-time ad-hoc or bleeding-edge analyses.

For the purposes of reproducibility, a pipeline can be represented as an atomic tool - with known dependencies, inputs, outputs, and parameters. A pipeline's constituent tools can be loosely represented by the dependency management formats discussed above, but many pipelines are designed to run on high-performance computing grids or clusters, the composition of which can be difficult to replicate locally. Describing a pipeline is more than merely an ordered collection of its parts. Maintaining metadata integrity throughout a workflow depends on compiling formatted metadata at the time of data collection and using those metadata as the primary configuration files for analytical pipelines. Reproducible research is best accomplished by building "metadata-driven analyses," whereby workflow engines can infer tasks from recorded metadata rather than relying on manual intervention.

Examples

CWL - A configuration-based framework for interoperability

The Common Workflow Language (CWL) ¹⁴⁷ 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 13). While this metadata is primarily operational in nature and rarely accessed outside the context of a compatible runner such as Rabix ¹⁴⁸ or Toil ¹⁴⁹, CWL also enables a 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 ⁶⁰, ELIXIR's bio.tools ⁵⁶ registry and Research Resource Identifiers (RRIDs) ¹⁵⁰. This wrapping of command line tool interfaces is used by GA4GH Dockstore ¹²⁹ 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 more amenable to being used as metadata for all components of an analysis, as demonstrated in the EOSC-Life [Workflow Hub](#), which allows workflows and scripts in any format to be accompanied by an *abstract* CWL that provide structural and semantic descriptions.

```
#!/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 standard operating procedure for genomic variant detection using GATK4.
  It is hoped to be effective and useful for getting SARS-CoV-2 genome variants.

  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 ]
```

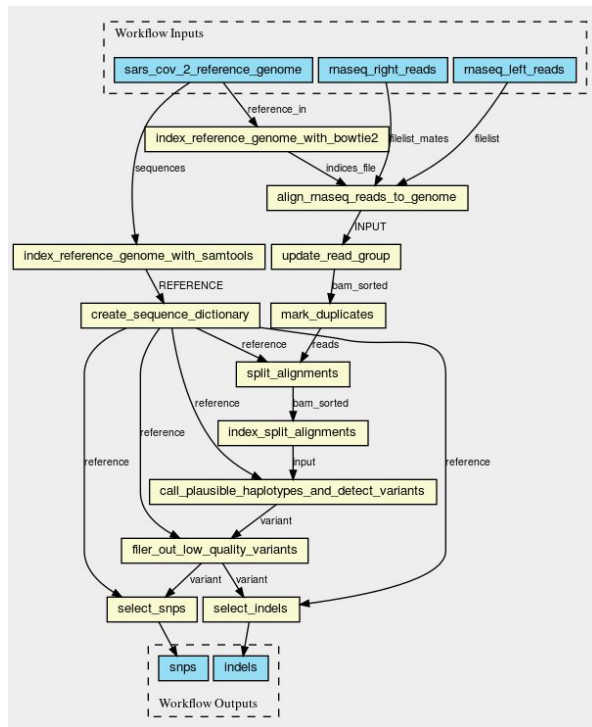


Figure 13: Snippets of a COVID-19 variant detection CWL workflow and the workflow as viewed through the cwl-viewer¹⁵¹. Note the EDAM file definitions.

Future directions

Interoperable script and workflow provenance

For future metadata to support pipeline reproducibility must accommodate 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.

Pimental et al. reviewed and categorized 27 approaches to collecting provenance from scripts¹⁵². 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 while four approaches - RDataTracker¹⁵³, SPADE¹⁵⁴, StarFlow¹⁵⁵, and YesWorkflow¹⁵⁶ - 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 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 which 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 retrospective¹⁵⁷. As part of the WINGS project, Garijo¹⁵⁸ uses 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 CWLProv⁷² 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 are under version 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 ¹⁵⁹. A YAML configuration file connects these parts, 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 ¹⁶⁰.

```
id: b9b0099e-9f8d-4a33-8acf-cb0c062efaec
spec_version: 1
main: workflow.Rmd
display: paper.html
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"
```

Figure 14: erc.yml example file, see the specification at <https://o2r.info/erc-spec/>.

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 are Research Objects ¹⁶¹ and the subsequent RO-Crate ¹⁶² projects, 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) which 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.

```
{
  "@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": "2017",
      "name": "Data files associated with the manuscript: Effects of facilitated family case conferencing for...",
      "description": "Palliative care planning for nursing home residents with advanced dementia...",
      "license": {
        "@id": "https://creativecommons.org/licenses/by-nc-sa/3.0/au/"
      }
    },
    {
      "@id": "https://creativecommons.org/licenses/by-nc-sa/3.0/au/",
      "@type": "CreativeWork",
      "description": "This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike...",
      "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)"
    }
  ]
}
```

Figure 15: RO-Crate metadata

An alternative approach to binding is to leverage existing work in "application profiles" ¹⁶³, a highly customizable means of combining namespaces from different metadata schemas. Application profiles follow along the Singapore Framework, and guidelines supported by the Dublin Core Metadata Initiative (DCMI).

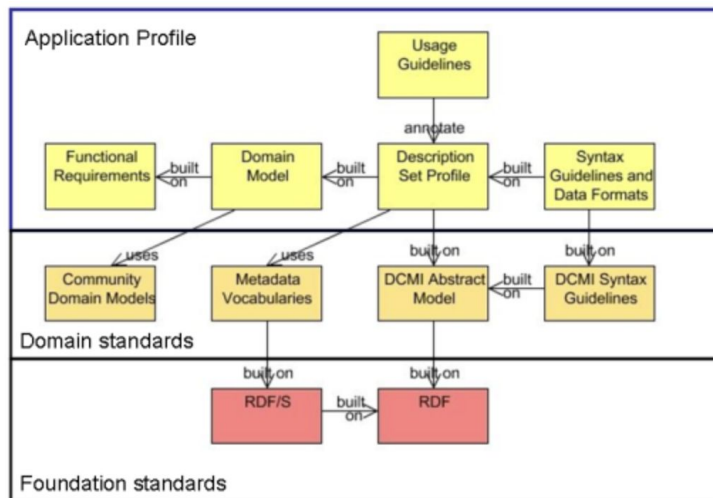


Figure 16: Singapore Framework application profile model

5. 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. Distinguishing this type of publication-level metadata from input metadata is not clear cut but

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. 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^{164,165} and recommendation¹⁶⁶ 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 ontology¹⁶⁷. While these efforts are noble, they can never entirely bridge the gap between human-readable protocols and RCR.

Journals share an important responsibility to enforce and incentivize reproducible research, 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. Some journals, such as Nature Methods, do require authors to submit source code¹⁶⁸. Of the most prestigious life science journals (Nature, Science, Cell), the requirements vary considerably and it is not clear how these guidelines are actually enforced¹⁶⁹

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. Computational provenance between all figures and tables in a manuscript and the underlying analysis is an open area of research that we discuss below.

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 ⁸⁵, SBML ¹⁷⁰, SBGN ¹⁷¹, BioPAX ¹⁷², GO-CAM ¹⁷³) have the facility to describe the interactions between biological entities that are often elucidated through laboratory or clinical experimentation. Further, there are several organizations ^{174–178} 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 ^{179,180} curation workflows based on NLP-based relation extraction systems ^{181–183} and assemblers ¹⁸⁴ 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-language.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 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, 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) ¹⁸⁵ namespace to describe the RAF, MEK, and ERK protein families.


```

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))

```

Figure 17: MAPK signalling pathway in Biological Expression Language (BEL)

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 ¹⁸⁶ was published with an inline live RMarkdown Binder analysis as part of a proof-of-concept of the publisher's Reproducible Document Stack (RDS) ¹⁸⁷. 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 to aid researchers in their daily activities, but also in sharing research (data, code, publications), and contributing to open science is extensive. If we promote metadata as the "glue" of reproducible research, what does that entail for the metadata and RCR communities? While there are overlapping standards, and no one metadata schema can support all aspects of the analytic stack, it is important to recognize that the metadata developments pursued, particularly the standards that are shared and maintained by a community, many of which have gone through formal standards review processes, demonstrate value to their communities. Science has no boundary, and while these standards may have been developed to meet more specific needs as part of the research life-cycle, as reviewed above, they have a continuing value for RCR.

In our review, we have attempted to describe metadata as it addresses reproducibility across the analytic stack. Two principal components: 1) Embeddedness vs connectedness and the 2)

methodology weight and standardization appear to be recurring themes across all RCR metadata facets.

Embeddedness vs 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. 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 ¹⁸⁸. The continued increase in the use of such imaging data has led to efforts to further leverage biomedical ontologies in tags ¹⁸⁹ and issue DOIs to individual images ¹⁹⁰. As discussed above, the lack of support for complex metadata structures has not significantly 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. In the case of Binary Alignment Map ¹⁹¹ files used to store genomic alignments, file-level metadata resides in 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 be 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 commas or semicolons 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 instruments 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 ¹⁹² and geospatial data ¹⁹³. RDF triples are the core of knowledge graph projects such as DBpedia ¹⁹⁴ and Bio2RDF ¹⁹⁵. 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 AI approaches ¹⁹⁶, has exposed the need for reproducibility and metadata standards in this area ¹⁹⁷.

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 ¹⁹⁸. This dependence on data and services hosted by others, however, introduces the threat of "workflow decay" ¹⁹⁹ that requires extensive provenance tracking to freeze inputs and tools in order to ensure reproducibility at a later date.

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 RCR is a two-sided coin. On one hand, as noted by Aranguren and Wilkinson ²⁰⁰, the semantic web provides a formalized means 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 ²⁰¹ 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.

They provide a common source reference point for several unrelated analyses, but that can serve as a critical point of failure should they disappear. Projects serving to provide long-term archival solutions for scientific analyses need to cache or download webservice data. Along the same lines, often studies are conducted entirely from dedicated databases - relational, so-called

"NoSQL" solutions - key-value, document stores, or column-stores. These can introduce substantial issues to portability and reproducibility, especially when studies access relational joins across subsets of these databases.

Methodology weight and standardization

Our review has spotlighted several metadata solutions across a spectrum of heavyweight vs lightweight solutions, bespoke vs 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) 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 RCR has been from the ground up, and often originally intended for purposes other than reproducibility. One could argue the vertical integration required to take raw data to report would enable small labs and individual investigators to control all aspects of the research process, designing pipelines and metadata standards for RCR. Most of these solutions, however, are "bespoke," or custom-designed to address the problem at hand. A good example is the tximeta Bioconductor package, which implements reference transcriptome provenance for RNA-Seq experiments, extending a number of popular transcript quantification tools with checksum-based tracking and identification²⁰². While this is an elegant solution, tximeta is focused on one analysis pattern.

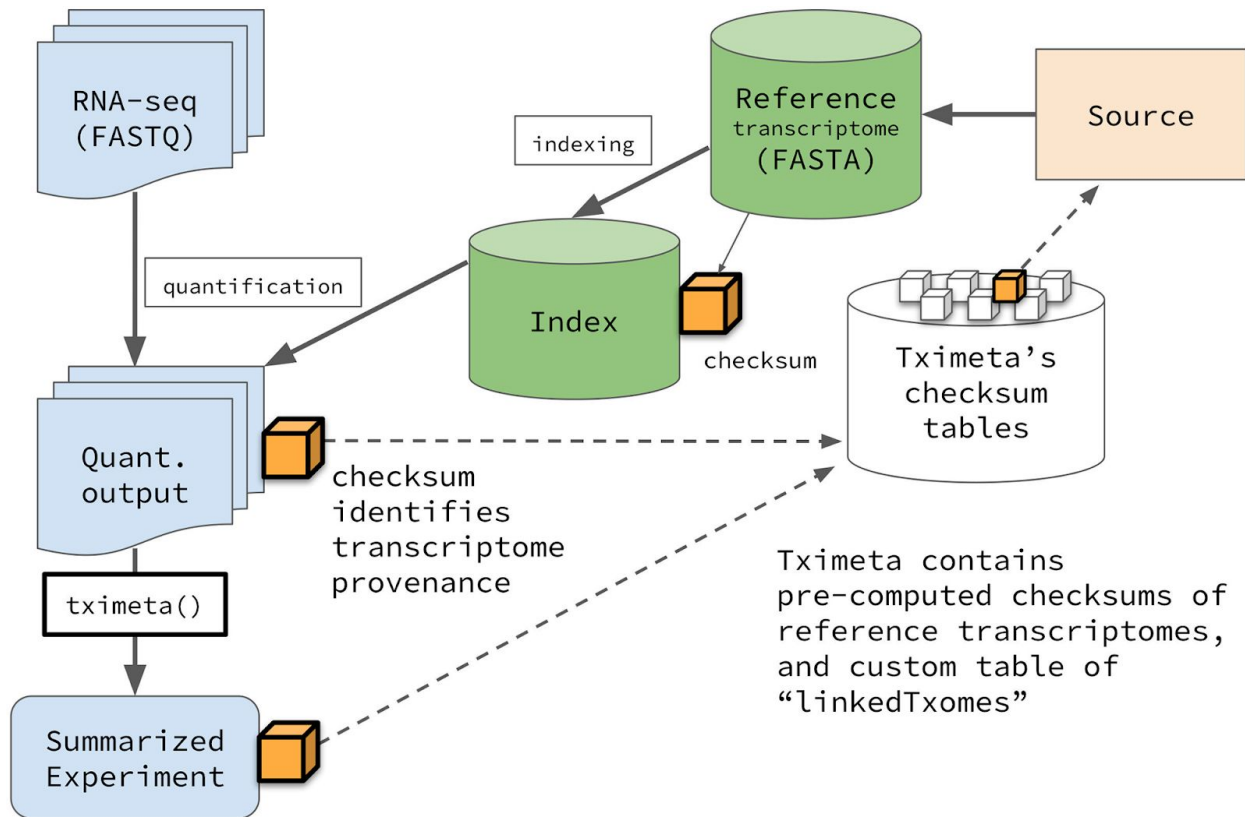


Figure 18: The high-level schematic of tximeta ²⁰²

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. In addition to the greater scientific community, a perhaps unexpected source of impetus has been from private companies offering forecasting services and predictions derived from machine learning and deep learning techniques for whom a lack of reproducibility is a legal liability. These data science customers have driven the development of a number of tools along the analytic stack (Docker, Airflow, Luigi, MLflow) that have been open-sourced and adopted scientific users.

Metadata capital and reuse

The term "metadata capital" ²⁰³ 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 RCR. 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 ¹⁴. Metadata reuse - for replication, generalization, meta-analyses, or general research use is enabled by RCR and elemental to FAIR. Reuse typically demands greater metadata needs than narrow-sense reproduction, for instance, to control for batch effects or various assumptions that go into original research ²⁰⁴. 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. Essential for reproducibility, surveys suggest provenance information is an important criteria for reuse in the physical sciences ²⁰⁵. Metadata for data reuse has relevance for data harmonization for biomedical applications, such as toward highly granular phenotype ontologies, genotype-phenotype meta-analyses ²⁰⁶, generating synthetic controls for clinical trials, and, consent metadata such as the Data Use Ontology ²⁰⁷ to describe allowed downstream usage of patient data. Designing metadata for the needs of general reuse, especially outside narrow scientific domains, requires greater foresight than that needed for RCR but authors can follow similar templates.

Recommendations & Future Work

Widespread adoption of RCR is highly dependent on a cultural shift within the scientific community ^{208,209} promoted by both journals and funding agencies. The allegorical "stick" of higher RCR standards should be accompanied by carrots in the form of publication incentives. One of these 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 data sets. An "advocated software peer review" enabled by RCR and conducted by reviewers taking a hands-on approach to strengthening analyses using collaborative interactive notebooks or other tools. ²¹⁰

One interesting development in this area 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 ²¹¹. 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.

In terms of the analytic stack, there are several areas which 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 ²¹², but generally Such terms could disambiguate concepts, point to the provenance of findings within a result section or from a figure, and accelerate linked data and discovery.

Secondly 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 users understanding of, for instance, existing technique relates most closely with a new one. This will ensure the broader goals of reproducibility.

Statistical metadata is vital for users to discover, and reviewers to evaluate complex statistical analyses²¹³, but metadata that describe statistical methods is largely non-existent. 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 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 subtle that even experienced data scientists may have to evaluate them empirically but may substantially change model predictions given limited data.

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. 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.

Finally there needs to be greater emphasis on translation between embedded and distributed metadata solutions. As discussed, files which 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 which 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 RCR.

Acknowledgements

We wish to thank Jian Qin and Farah Zaib Khan for their helpful comments and suggestions.

References

1. Margolis, R. *et al.* 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 (2014).
2. Brito, J. J. *et al.* *Recommendations to enhance rigor and reproducibility in biomedical research*. https://mangul-lab-usc.github.io/enhancing_reproducibility/ (2020).
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; Committee on Reproducibility and Replicability in Science. *Reproducibility and Replicability in Science*. (National Academies Press (US), 2019).
4. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data* **3**, 160018 (2016).
5. Donoho, D. L. An invitation to reproducible computational research. *Biostatistics* **11**, 385–388 (2010).

6. Stodden, V., Borwein, J. & Bailey, D. H. Setting the default to reproducible. *computational science research. SIAM News* **46**, 4–6 (2013).
7. Whitaker, K. Showing your working: A guide to reproducible neuroimaging analyses. *figshare* (2016) doi:10.6084/m9.figshare.4244996.v1.
8. The Turing Way Community *et al.* *The Turing Way: A Handbook for Reproducible Data Science*. (2019). doi:10.5281/zenodo.3233986.
9. Barba, L. A. Terminologies for Reproducible Research. *arXiv [cs.DL]* (2018).
10. Baggerly, K. Disclose all data in publications. *Nature* **467**, 401 (2010).
11. Begley, C. G. & Ellis, L. M. Drug development: Raise standards for preclinical cancer research. *Nature* **483**, 531–533 (2012).
12. Ioannidis, J. P. A. *et al.* Repeatability of published microarray gene expression analyses. *Nat. Genet.* **41**, 149–155 (2009).
13. Motulsky, H. J. Common misconceptions about data analysis and statistics. *J. Pharmacol. Exp. Ther.* **351**, 200–205 (2014).
14. Baker, M. 1,500 scientists lift the lid on reproducibility. *Nature News* **533**, 452 (2016).
15. Ioannidis, J. P. A. Why most published research findings are false. *PLoS Med.* **2**, e124 (2005).
16. Fanelli, D. Opinion: Is science really facing a reproducibility crisis, and do we need it to? *Proc. Natl. Acad. Sci. U. S. A.* **115**, 2628–2631 (2018).
17. Leipzig, J. *Awesome Reproducible Research*. (2019). doi:10.5281/zenodo.3564746.
18. Lehrer, J. The truth wears off. *New Yorker* **13**, 229 (2010).
19. Greenberg, J., Swauger, S. & Feinstein, E. Metadata capital in a data repository. in *International Conference on Dublin Core and Metadata Applications* 140–150 (2013).
20. Rousidis, D., Garoufallou, E., Balatsoukas, P. & Sicilia, M.-A. Metadata for Big Data: a

- preliminary investigation of metadata quality issues in research data repositories. *Inf. Serv. Use* **34**, 279–286 (2014).
21. Ekbja, H. *et al.* Big data, bigger dilemmas: A critical review. *J Assn Inf Sci Tec* **66**, 1523–1545 (2015).
 22. Warden, P. The Machine Learning Reproducibility Crisis. *Pete Warden's blog* <https://petewarden.com/2018/03/19/the-machine-learning-reproducibility-crisis/> (2018).
 23. Bouthillier, X., Laurent, C. & Vincent, P. Unreproducible Research is Reproducible. in *Proceedings of the 36th International Conference on Machine Learning* (eds. Chaudhuri, K. & Salakhutdinov, R.) vol. 97 725–734 (PMLR, 2019).
 24. Schelter, S., Boese, J.-H., Kirschnick, J., Klein, T. & Seufert, S. Automatically tracking metadata and provenance of machine learning experiments. in *Machine Learning Systems Workshop at NIPS* 27–29 (2017).
 25. Obels, P., Lakens, D., Coles, N. A., Gottfried, J. & Green, S. A. Analysis of Open Data and Computational Reproducibility in Registered Reports in Psychology. (2019)
doi:10.31234/osf.io/fk8vh.
 26. Rauh, S. *et al.* Reproducible and Transparent Research Practices in Published Neurology Research. doi:10.1101/763730.
 27. Stodden, V., Krafczyk, M. S. & Bhaskar, A. 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* 1–5 (Association for Computing Machinery, 2018).
 28. Stagge, J. H. *et al.* Assessing data availability and research reproducibility in hydrology and water resources. *Sci Data* **6**, 190030 (2019).
 29. Nüst, D. *et al.* Reproducible research and GIScience: an evaluation using AGILE

- conference papers. *PeerJ* **6**, e5072 (2018).
30. Sandve, G. K., Nekrutenko, A., Taylor, J. & Hovig, E. Ten simple rules for reproducible computational research. *PLoS Comput. Biol.* **9**, e1003285 (2013).
 31. Collberg, C. *et al.* Measuring reproducibility in computer systems research. *Department of Computer Science, University of Arizona, Tech. Rep* (2014).
 32. Piccolo, S. R. & Frampton, M. B. Tools and techniques for computational reproducibility. *Gigascience* **5**, 30 (2016).
 33. FitzJohn, R., Pennell, M., Zanne, A. & Cornwell, W. Reproducible research is still a challenge. *rOpenSci*. 2014.
 34. Ball, A. Scientific data application profile scoping study report. *June 3rd* (2009).
 35. Ball, A., Greenberg, J., Jeffery, K. & Koskela, R. RDA Metadata Standards Directory Working Group. (2016).
 36. Riley, J. Understanding metadata. *Washington DC, United States: National Information Standards Organization* (<http://www.niso.org/publications/press/UnderstandingMetadata.pdf>) 23 (2017).
 37. Qin, J. & Zeng, M. *Metadata*. (ALA-Neal Schuman, 2016).
 38. Brazma, A. *et al.* Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat. Genet.* **29**, 365–371 (2001).
 39. Ashburner, M. *et al.* Gene Ontology: tool for the unification of biology. *Nat. Genet.* **25**, 25–29 (2000).
 40. Pearson, W. R. [5] Rapid and sensitive sequence comparison with FASTP and FASTA. in *Methods in Enzymology* vol. 183 63–98 (Academic Press, 1990).
 41. Wilkinson, M. D. *et al.* A design framework and exemplar metrics for FAIRness. *bioRxiv* 225490 (2017) doi:10.1101/225490.

42. Paskin, N. Digital object identifier (DOI®) system. *Encyclopedia of library and information sciences* **3**, 1586–1592 (2010).
43. Sansone, S.-A. *et al.* FAIRsharing as a community approach to standards, repositories and policies. *Nat. Biotechnol.* **37**, 358–367 (2019).
44. Qin, J., Dobreski, B. & Brown, D. Metadata and Reproducibility: A Case Study of Gravitational Wave Research Data Management. *International Journal of Digital Curation* **11**, 218–231 (2016).
45. Page, K. *et al.* From workflows to Research Objects: an architecture for preserving the semantics of science. in *Proceedings of the 2nd International Workshop on Linked Science* (pdfs.semanticscholar.org, 2012).
46. Lenhardt, W., Ahalt, S., Blanton, B., Christopherson, L. & Idaszak, R. Data management lifecycle and software lifecycle management in the context of conducting science. *Journal of Open Research Software* **2**, (2014).
47. Michener, W. K. Meta-information concepts for ecological data management. *Ecol. Inform.* **1**, 3–7 (2006/1).
48. Bidgood, W. D., Jr & Horii, S. C. Introduction to the ACR-NEMA DICOM standard. *Radiographics* **12**, 345–355 (1992).
49. Robertson, T. *et al.* The GBIF integrated publishing toolkit: facilitating the efficient publishing of biodiversity data on the internet. *PLoS One* **9**, e102623 (2014).
50. Bernstein, H. J. *et al.* Specification of the Crystallographic Information File format, version 2.0. *J. Appl. Crystallogr.* **49**, 277–284 (2016).
51. Chirico, R. D., Frenkel, M., Diky, V. V., Marsh, K. N. & Wilhoit, R. C. 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 (2003).
52. Cuellar, A. A. *et al.* An Overview of CellML 1.1, a Biological Model Description Language. *Simulation* **79**, 740–747 (2003).
 53. Alter, G., Gonzalez-Beltran, A., Ohno-Machado, L. & Rocca-Serra, P. The Data Tags Suite (DATS) model for discovering data access and use requirements. *Gigascience* **9**, (2020).
 54. Andersson, L. *et al.* Coordinated international action to accelerate genome-to-phenome with FAANG, the Functional Annotation of Animal Genomes project. *Genome Biol.* **16**, 57 (2015).
 55. ISO/TC 276 - Biotechnology. ISO <https://www.iso.org/committee/4514241.html> (2020).
 56. Ison, J. *et al.* Tools and data services registry: a community effort to document bioinformatics resources. *Nucleic Acids Res.* **44**, D38–47 (2016).
 57. 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>.
 58. Dumbill, E. DOAP: Description of a Project. URL <http://trac.usefulinc.com/doap> (2010).
 59. Gil, Y., Ratnakar, V. & Garijo, D. OntoSoft: Capturing Scientific Software Metadata. in *Proceedings of the 8th International Conference on Knowledge Capture* 32 (ACM, 2015).
 60. Ison, J. *et al.* EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats. *Bioinformatics* **29**, 1325–1332 (2013).
 61. Malone, J. *et al.* The Software Ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation. *J. Biomed. Semantics* **5**, 25 (2014).
 62. Zheng, J. *et al.* The Ontology of Biological and Clinical Statistics (OBCS) for standardized and reproducible statistical analysis. *J. Biomed. Semantics* **7**, 53 (2016).
 63. STATO: an Ontology of Statistical Methods. <http://stato-ontology.org/>.
 64. Capadisli, S., Auer, S. & Ngonga Ngomo, A.-C. Linked SDMX data. *Semantic Web* **6**,

- 105–112 (2015).
65. Hoyle, L. & Wackerow, J. DDI as a Common Format for Export and Import for Statistical Packages. (2016).
 66. Esteves, D. *et al.* MEX vocabulary: a lightweight interchange format for machine learning experiments. in *Proceedings of the 11th International Conference on Semantic Systems* 169–176 (Association for Computing Machinery, 2015).
 67. Publio, G. C. *et al.* ML-Schema: Exposing the Semantics of Machine Learning with Schemas and Ontologies. *arXiv [cs.LG]* (2018).
 68. Peter, A. *et al.* Common Workflow Language, v1.0. *figshare* (2016)
doi:10.6084/m9.figshare.3115156.v2.
 69. Santana-Perez, I. *et al.* Reproducibility of execution environments in computational science using Semantics and Clouds. *Future Gener. Comput. Syst.* **67**, 354–367 (2017/2).
 70. Moreau, L. *et al.* Open Provenance Model (OPM) OWL Specification, October 2010. *URL* <http://openprovenance.org/model/opmo>. (Cited on page 139.).
 71. Cheney, J., Corsar, D., Garijo, D. & Soiland-Reyes, S. Prov-o: The prov ontology. *W3C Working Draft* (2012).
 72. Khan, F. Z. *et al.* Sharing interoperable workflow provenance: A review of best practices and their practical application in CWLProv. *Gigascience* **8**, (2019).
 73. Cao, Y. *et al.* ProvONE: extending PROV to support the DataONE scientific community. *homepages.cs.ncl.ac.uk*.
 74. Ciccarese, P. *et al.* PAV ontology: provenance, authoring and versioning. *J. Biomed. Semantics* **4**, 37 (2013).
 75. Kunze, J., Littman, J., Madden, E., Scancella, J. & Adams, C. The BagIt File Packaging Format (V1.0). (2018) doi:10.17487/rfc8493.

76. Sefton¹, P., Carragáin, E. Ó., Goble, C. & Soiland-Reyes, S. Introducing RO-Crate: research object data packaging. *conference.eresearch.edu.au*.
77. Alterovitz, G. *et al.* Enabling precision medicine via standard communication of HTS provenance, analysis, and results. *PLoS Biol.* **16**, e3000099 (2018).
78. Weibel, S., Kunze, J., Lagoze, C. & Wolf, M. Dublin core metadata for resource discovery. *Internet Engineering Task Force RFC* **2413**, 132 (1998).
79. Journal Article Tag Suite 1.0: National Information Standards Organization standard of journal extensible markup language. *Sci. Educ.* doi:10.6087/kcse.2014.1.99.
80. Needleman, M. H. ONIX (Online Information Exchange). *Serials Review* **27**, 102–104 (2001).
81. Lipscomb, C. E. Medical Subject Headings (MeSH). *Bull. Med. Libr. Assoc.* **88**, 265–266 (2000).
82. Chan, L. M. *Library of Congress Subject Headings: Principles and Application. Third Edition.* (Libraries Unlimited, Inc., P.O. Box 6633, Englewood, CO 80155-6633 (paperback: ISBN-1-56308-191-1, \$35; clothbound: ISBN-1-56308-195-4, \$46)., 1995).
83. Clark, T., Ciccarese, P. N. & Goble, C. A. Micropublications: a semantic model for claims, evidence, arguments and annotations in biomedical communications. *J. Biomed. Semantics* **5**, 28 (2014).
84. Williams, A. J. *et al.* Open PHACTS: semantic interoperability for drug discovery. *Drug Discov. Today* **17**, 1188–1198 (2012).
85. Slater, T. Recent advances in modeling languages for pathway maps and computable biological networks. *Drug Discov. Today* **19**, 193–198 (2014).
86. Ciccarese, P. *et al.* The SWAN biomedical discourse ontology. *J. Biomed. Inform.* **41**, 739–751 (2008).

87. Peroni, S. The Semantic Publishing and Referencing Ontologies. in *Semantic Web Technologies and Legal Scholarly Publishing* (ed. Peroni, S.) 121–193 (Springer International Publishing, 2014).
88. Gangemi, A., Peroni, S., Shotton, D. & Vitali, F. The Publishing Workflow Ontology (PWO). *Semantic Web* 1–16 (2017).
89. Smith, B. *et al.* The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.* **25**, 1251–1255 (2007).
90. Graham, R. N. J., Perriss, R. W. & Scarsbrook, A. F. DICOM demystified: a review of digital file formats and their use in radiological practice. *Clin. Radiol.* **60**, 1133–1140 (2005).
91. Whitcher, B., Schmid, V. J. & Thornton, A. Working with the DICOM and NIFTI Data Standards in R. *J. Stat. Softw.* (2011) doi:10.18637/jss.v044.i06.
92. Guelde, M. O. *et al.* Quality of DICOM header information for image categorization. in *Medical Imaging 2002: PACS and Integrated Medical Information Systems: Design and Evaluation* vol. 4685 280–287 (International Society for Optics and Photonics, 2002).
93. Fedorov, A. *et al.* 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magn. Reson. Imaging* **30**, 1323–1341 (2012).
94. Herz, C. *et al.* dcmqi: An Open Source Library for Standardized Communication of Quantitative Image Analysis Results Using DICOM. *Cancer Res.* **77**, e87–e90 (2017).
95. Halpern, B., Frazier, M., Potapenko, J., Casey, K. & Koenig, K. Cumulative human impacts: Supplementary data. (2015).
96. Faith, J. J. *et al.* Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata. *Nucleic Acids Res.* **36**, D866–70 (2008).
97. Ramasamy, A., Mondry, A., Holmes, C. C. & Altman, D. G. Key issues in conducting a meta-analysis of gene expression microarray datasets. *PLoS Med.* **5**, e184 (2008).

98. Rocca-Serra, P. *et al.* ISA software suite: supporting standards-compliant experimental annotation and enabling curation at the community level. *Bioinformatics* **26**, 2354–2356 (2010).
99. Pezoa, F., Reutter, J. L., Suarez, F., Ugarte, M. & Vrgoč, D. Foundations of JSON Schema. in *Proceedings of the 25th International Conference on World Wide Web* 263–273 (International World Wide Web Conferences Steering Committee, 2016).
100. Janowicz, K. *et al.* Five stars of linked data vocabulary use. *Semantic Web* **5**, 173–176 (2014).
101. Brickley, D., Burgess, M. & Noy, N. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. in *The World Wide Web Conference* 1365–1375 (Association for Computing Machinery, 2019).
102. Oberkamp, H., Krieg, H., Senger, C., Weber, T. & Colsman, W. 20 Allotrope Data Format--Semantic Data Management in Life Sciences. pdf. (2018).
103. Stathias, V. *et al.* Sustainable data and metadata management at the BD2K-LINCS Data Coordination and Integration Center. *Sci Data* **5**, 180117 (2018).
104. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**, 550 (2014).
105. Beaulieu-Jones, B. K. & Greene, C. S. Reproducible Computational Workflows with Continuous Analysis. *bioRxiv* 056473 (2016) doi:10.1101/056473.
106. Palmblad, M., Lamprecht, A.-L., Ison, J. & Schwämmle, V. Automated workflow composition in mass spectrometry-based proteomics. *Bioinformatics* **35**, 656–664 (2019).
107. Hillion, K.-H. *et al.* Using bio.tools to generate and annotate workbench tool descriptions. *F1000Res.* **6**, (2017).
108. Bedo, J. Bioshake: a Haskell EDSL for bioinformatics pipelines. *bioRxiv* 529479 (2019)

doi:10.1101/529479.

109. Amstutz, P. *et al.* Portable workflow and tool descriptions with the CWL. in *Bioinformatics Open Source Conference* (2015).
110. Kumar, A., Grüning, B. & Backofen, R. Tool recommender system in Galaxy using deep learning. *bioRxiv* 838599 (2019) doi:10.1101/838599.
111. Jones, M. B. *et al.* CodeMeta: an exchange schema for software metadata. KNB Data Repository. (2016) doi:10.5063/schema/codemeta-1.0.
112. *baydem*. (Github).
113. Smith, A. M., Katz, D. S. & Niemeyer, K. E. Software citation principles. *PeerJ Comput. Sci.* **2**, e86 (2016).
114. Wattanakriengkrai, S. *et al.* GitHub Repositories with Links to Academic Papers: Open Access, Traceability, and Evolution. *arXiv [cs.SE]* (2020).
115. Dozmorov, M. GitHub statistics as a measure of the impact of open-source bioinformatics software. *Frontiers in Bioengineering and Biotechnology* **6**, 198 (2018).
116. Pimentel, J. F., Murta, L., Braganholo, V. & Freire, J. A Large-Scale Study About Quality and Reproducibility of Jupyter Notebooks. in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)* 507–517 (2019).
117. Boettiger, C. An Introduction to Docker for Reproducible Research. *Oper. Syst. Rev.* **49**, 71–79 (2015).
118. Grüning, B. *et al.* Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat. Methods* **15**, 475–476 (2018).
119. ENCODE Project Consortium *et al.* Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature* **447**, 799–816 (2007).
120. Hung, L.-H. *et al.* Building containerized workflows for RNA-seq data using the

- BioDepot-workflow-Builder (BwB). *bioRxiv* 099010 (2017) doi:10.1101/099010.
121. Moreews, F. *et al.* BioShaDock: a community driven bioinformatics shared Docker-based tools registry. *F1000Res.* **4**, 1443 (2015).
 122. Belmann, P. *et al.* Bioboxes: standardised containers for interchangeable bioinformatics software. *Gigascience* **4**, 47 (2015).
 123. da Veiga Leprevost, F. *et al.* BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics* **33**, 2580–2582 (2017).
 124. Allamanis, M. & Sutton, C. Mining Source Code Repositories at Massive Scale Using Language Modeling. in *Proceedings of the 10th Working Conference on Mining Software Repositories* 207–216 (IEEE Press, 2013).
 125. Gentleman, R. C. *et al.* Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.* **5**, R80 (2004).
 126. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
 127. Tierney, N. J. & Ram, K. A Realistic Guide to Making Data Available Alongside Code to Improve Reproducibility. *arXiv [cs.DL]* (2020).
 128. Open Containers Initiative. *Open Containers Initiative* <https://www.opencontainers.org/>.
 129. Yuen, D. *et al.* *Ga4Gh/Dockstore: 1.0.* (Zenodo, 2016). doi:10.5281/zenodo.154185.
 130. Leisch, F. Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis. in *Compstat* (eds. Härdle, P. D. W. & Rönz, P. D. B.) 575–580 (Physica-Verlag HD, 2002).
 131. Xie, Y. knitr: a comprehensive tool for reproducible research in R. *Implement Reprod Res* **1**, 20 (2014).
 132. Team, R. & Others. RStudio: integrated development for R. *RStudio, Inc. , Boston, MA URL* <http://www.rstudio.com> **42**, 14 (2015).

133. Kluyver, T. *et al.* Jupyter Notebooks—a publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas* 87 (2016).
134. Shen, H. Interactive notebooks: Sharing the code. *Nature* **515**, 151–152 (2014).
135. Zhang, S., Zhang, C. & Yang, Q. Data preparation for data mining. *Appl. Artif. Intell.* **17**, 375–381 (2003).
136. Rosenberg, D. M. & Horn, C. C. Neurophysiological analytics for all! Free open-source software tools for documenting, analyzing, visualizing, and sharing using electronic notebooks. *J. Neurophysiol.* **116**, 252–262 (2016).
137. Jupyter, P. *et al.* Binder 2.0-Reproducible, interactive, sharable environments for science at scale. in *Proceedings of the 17th python in science conference* vol. 113 120 (2018).
138. Allaire, J. J., Wickham, H., Xie, Y., Vaidyanathan, R. & Others. Ricles: Article Formats for R Markdown. (2016).
139. Pineau, J. *et al.* Improving Reproducibility in Machine Learning Research (A Report from the NeurIPS 2019 Reproducibility Program). *arXiv [cs.LG]* (2020).
140. Ćwiek-Kupczyńska, H. *et al.* Semantic concept schema of the linear mixed model of experimental observations. *Sci Data* **7**, 70 (2020).
141. Zaharia, M. *et al.* Accelerating the Machine Learning Lifecycle with MLflow. *IEEE Data Eng. Bull.* **41**, 39–45 (2018).
142. Leipzig, J. A review of bioinformatic pipeline frameworks. *Brief. Bioinform.* (2016) doi:10.1093/bib/bbw020.
143. Goecks, J., Nekrutenko, A., Taylor, J. & Galaxy Team. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* **11**, R86 (2010).

144. Altintas, I. *et al.* Kepler: an extensible system for design and execution of scientific workflows. in *Proceedings. 16th International Conference on Scientific and Statistical Database Management, 2004*. 423–424 (ieeexplore.ieee.org, 2004).
145. Berthold, M. R. *et al.* KNIME - the Konstanz Information Miner: Version 2.0 and Beyond. *SIGKDD Explor. Newsl.* **11**, 26–31 (2009).
146. Hull, D. *et al.* Taverna: a tool for building and running workflows of services. *Nucleic Acids Res.* **34**, W729–32 (2006).
147. Peter, A. *et al.* Common Workflow Language, v1.0. *figshare* (2016)
doi:10.6084/m9.figshare.3115156.v2.
148. Kaushik, G. *et al.* RABIX: AN OPEN-SOURCE WORKFLOW EXECUTOR SUPPORTING RECOMPUTABILITY AND INTEROPERABILITY OF WORKFLOW DESCRIPTIONS. *Pac. Symp. Biocomput.* **22**, 154–165 (2016).
149. Vivian, J. *et al.* Toil enables reproducible, open source, big biomedical data analyses. *Nat. Biotechnol.* **35**, 314–316 (2017).
150. Bandrowski, A. E. & Martone, M. E. RRIDs: A Simple Step toward Improving Reproducibility through Rigor and Transparency of Experimental Methods. *Neuron* **90**, 434–436 (2016).
151. Robinson, M., Soiland-Reyes, S., Crusoe, M. R. & Goble, C. CWL Viewer: The Common Workflow Language Viewer. in *Bioinformatics Open Source Conference (BOSC) 2017* (2017).
152. Pimentel, J. F., Freire, J., Murta, L. & Braganholo, V. A Survey on Collecting, Managing, and Analyzing Provenance from Scripts. *ACM Computing Surveys* vol. 52 1–38 (2019).
153. Lerner, B. & Boose, E. RDataTracker: collecting provenance in an interactive scripting environment. in *6th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2014)* (2014).

154. Gehani, A., Kazmi, H. & Irshad, H. Scaling SPADE to 'Big Provenance'. in *TaPP* (2016).
155. Angelino, E., Yamins, D. & Seltzer, M. StarFlow: A Script-Centric Data Analysis Environment. *Lecture Notes in Computer Science* 236–250 (2010)
doi:10.1007/978-3-642-17819-1_27.
156. McPhillips, T. *et al.* YesWorkflow: A User-Oriented, Language-Independent Tool for Recovering Workflow Information from Scripts. *arXiv [cs.SE]* (2015).
157. Freire, J. Making Computations and Publications Reproducible with VisTrails. *Comput. Sci. Eng.* **14**, 18–25 (2012).
158. Garijo, D., Gil, Y. & Corcho, O. Abstract, link, publish, exploit: An end to end framework for workflow sharing. *Future Gener. Comput. Syst.* **75**, 271–283 (2017).
159. Nüst, D. *et al.* Opening the publication process with executable research compendia. *D-Lib Magazine* **23**, (2017).
160. Konkol, M., Kray, C. & Suleiman, J. Creating Interactive Scientific Publications using Bindings. *Proc. ACM Hum.-Comput. Interact.* **3**, 1–18 (2019).
161. Bechhofer, S. *et al.* Why linked data is not enough for scientists. *Future Gener. Comput. Syst.* **29**, 599–611 (2013/2).
162. Carragáin, E. Ó., Goble, C., Sefton, P. & Soiland-Reyes, S. A lightweight approach to research object data packaging. in *Bioinformatics Open Source Conference (BOSC) 2019* (2019).
163. Heery, R. & Patel, M. Application Profiles: Mixing and Matching Metadata Schemas. *Ariadne* (2000).
164. Duck, G., Nenadic, G., Brass, A., Robertson, D. L. & Stevens, R. Extracting patterns of database and software usage from the bioinformatics literature. *Bioinformatics* **30**, i601–8 (2014).

165. Eales, J. M., Pinney, J. W., Stevens, R. D. & Robertson, D. L. Methodology capture: discriminating between the 'best' and the rest of community practice. *BMC Bioinformatics* **9**, 359 (2008).
166. Halioui, A., Valtchev, P. & Diallo, A. B. Towards an ontology-based recommender system for relevant bioinformatics workflows. *bioRxiv* 082776 (2016) doi:10.1101/082776.
167. Sahoo, S. S., Valdez, J., Kim, M., Rueschman, M. & Redline, S. ProvCaRe: Characterizing scientific reproducibility of biomedical research studies using semantic provenance metadata. *Int. J. Med. Inform.* **121**, 10–18 (2019).
168. Evanko, D. Guidelines for algorithms and software in Nature Methods : Methagora. <http://blogs.nature.com/methagora/2014/02/guidelines-for-algorithms-and-software-in-nature-methods.html>.
169. Ince, D. C., Hatton, L. & Graham-Cumming, J. The case for open computer programs. *Nature* **482**, 485–488 (2012).
170. Hucka, M. *et al.* The Systems Biology Markup Language (SBML): language specification for level 3 version 2 core. *J. Integr. Bioinform.* **15**, (2018).
171. Le Novère, N. *et al.* The systems biology graphical notation. *Nat. Biotechnol.* **27**, 735 (2009).
172. Demir, E. *et al.* The BioPAX community standard for pathway data sharing. *Nat. Biotechnol.* **28**, 1308 (2010).
173. The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Res.* **47**, D330–D338 (2019).
174. Cerami, E. G. *et al.* Pathway Commons, a web resource for biological pathway data. *Nucleic Acids Res.* **39**, 685 (2011).
175. Fabregat, A. *et al.* The Reactome pathway knowledgebase. *Nucleic Acids Res.* **46**, D649

- (2018).
- 176.Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y. & Morishima, K. KEGG: New perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* **45**, D353 (2017).
- 177.Perfetto, L. *et al.* SIGNOR: A database of causal relationships between biological entities. *Nucleic Acids Res.* **44**, D548 (2016).
- 178.Slenter, D. N. *et al.* WikiPathways: a multifaceted pathway database bridging metabolomics to other omics research. *Nucleic Acids Res.* **46**, D661 (2018).
- 179.Hoyt, C. T. *et al.* Re-curation and rational enrichment of knowledge graphs in Biological Expression Language. *Database* **2019**, (2019).
- 180.Madan, S. *et al.* The BEL information extraction workflow (BELIEF): evaluation in the BioCreative V BEL and IAT track. *Database* **2016**, 1–17 (2016).
- 181.Allen, J. F., Swift, M. & De Beaumont, W. Deep semantic analysis of text. *Proceedings of the 2008 Conference on Semantics in Text Processing* **1**, 343 (2008).
- 182.McDonald, D. D. Issues in the Representation of Real Texts: The Design of Krisp. *Natural Language Processing and Knowledge Representation* **77** (2000).
- 183.Valenzuela-Escárcega, M. A. *et al.* Large-scale automated machine reading discovers new cancer-driving mechanisms. *Database* **2018**, 1 (2018).
- 184.Gyori, B. M. *et al.* From word models to executable models of signaling networks using automated assembly. *Mol. Syst. Biol.* **13**, 954 (2017).
- 185.Bachman, J. A., Gyori, B. M. & Sorger, P. K. FamPlex: A resource for entity recognition and relationship resolution of human protein families and complexes in biomedical text mining. *BMC Bioinformatics* **19**, 1–14 (2018).
- 186.Maciocci, G., Aufreiter, M. & Bentley, N. Introducing eLife's first computationally

- reproducible article. *eLife Labs [Internet]* **20**, (2019).
187. Aufreiter, M. & Penfold, N. *The Reproducible Document Stack reinvents the journal publication for a world of computationally reproducible research*. (2018).
doi:10.5281/zenodo.1311612.
188. Prior, F. *et al.* The public cancer radiology imaging collections of The Cancer Imaging Archive. *Sci Data* **4**, 170124 (2017).
189. Pérez, W., Tello, A., Saquicela, V., Vidal, M. & La Cruz, A. 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)* 2551–2554 (2015).
190. Bourne, P. E. DOIs for DICOM raw images: enabling science reproducibility. *Radiology* **275**, 3–4 (2015).
191. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).
192. Queralt-Rosinach, N., Piñero, J., Bravo, À., Sanz, F. & Furlong, L. I. DisGeNET-RDF: harnessing the innovative power of the Semantic Web to explore the genetic basis of diseases. *Bioinformatics* **32**, 2236–2238 (2016).
193. Geospatial Semantic Web. in *International Encyclopedia of Geography: People, the Earth, Environment and Technology* (eds. Richardson, D. *et al.*) vol. 284 1–6 (John Wiley & Sons, Ltd, 2016).
194. Auer, S. *et al.* DBpedia: A Nucleus for a Web of Open Data. in *The Semantic Web* 722–735 (Springer Berlin Heidelberg, 2007).
195. Dumontier, M. *et al.* 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* vol. 1272 401–404 (2014).
196. Kulmanov, M., Smaili, F. Z., Gao, X. & Hoehndorf, R. Machine learning with biomedical ontologies. *bioRxiv* 2020.05.07.082164 (2020) doi:10.1101/2020.05.07.082164.
197. Ali, M., Jabeen, H., Hoyt, C. T. & Lehman, J. The KEEN Universe: An Ecosystem for Knowledge Graph Embeddings with a Focus on Reproducibility and Transferability. *arXiv [cs.LG]* (2020).
198. Stein, L. D. The case for cloud computing in genome informatics. *Genome Biol.* **11**, 207 (2010).
199. De Roure, D. *et al.* Towards the preservation of scientific workflows. in *Procs. of the 8th International Conference on Preservation of Digital Objects (iPRES 2011)*. ACM (amiga.iaa.csic.es, 2011).
200. Aranguren, M. E. & Wilkinson, M. D. Enhanced reproducibility of SADI web service workflows with Galaxy and Docker. *Gigascience* **4**, 59 (2015).
201. Frey, J. G. & Bird, C. L. Cheminformatics and the Semantic Web: adding value with linked data and enhanced provenance. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **3**, 465–481 (2013).
202. Love, M. I. *et al.* Tximeta: Reference sequence checksums for provenance identification in RNA-seq. *PLoS Comput. Biol.* **16**, e1007664 (2020).
203. Greenberg, J. Big Metadata, Smart Metadata, and Metadata Capital: Toward Greater Synergy Between Data Science and Metadata. *Journal of Data and Information Science* **2**, 193 (2017).
204. Wang, H. & Webster, K. 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* (2019).

205. Murillo, A. P. Examining data sharing and data reuse in the DataONE environment. *Proceedings of the American Society for Information Science and Technology* **51**, 1–5 (2014).
206. Bernstein, M. N., Doan, A. & Dewey, C. N. MetaSRA: normalized human sample-specific metadata for the Sequence Read Archive. *Bioinformatics* **33**, 2914–2923 (2017).
207. *DUO*. (Github).
208. LeVeque, R. J., Mitchell, I. M. & Stodden, V. Reproducible research for scientific computing: Tools and strategies for changing the culture. *Computing in Science and Engineering* **14**, 13 (2012).
209. Arabas, S. *et al.* Case Studies and Challenges in Reproducibility in the Computational Sciences. *arXiv [cs.CE]* (2014).
210. NICTA Optimisation Research Group. NICTA-ORG/MLG Seminar: C. Titus Brown - Openness and Reproducibility in Computational Science. <https://www.youtube.com/watch?v=12hpAYr5ls0> (2014).
211. Clark, D. J. B., Wang, L., Jones, A. & Wojciechowicz, M. L. FAIRshake: toolkit to evaluate the findability, accessibility, interoperability, and reusability of research digital resources. *BioRxiv* (2019).
212. Himmelstein, D. S. *et al.* Open collaborative writing with Manubot. *PLoS Comput. Biol.* **15**, e1007128 (2019).
213. Dippo, C. S. *et al.* The Role of Metadata in Statistics.